ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

WEST UKRAINIAN NATIONAL UNIVERSITY

Комп'ютерна Інженерія

IV ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ СТУДЕНТІВ, АСПІРАНТІВ ТА МОЛОДИХ ВЧЕНИХ «ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ»

ІКСМ весна 2026

20 ТРАВНЯ 2026

Наукове товариство ЗУНУ
студентів, аспірантів, докторантів та молодих вчених

KI.WUNU.EDU.UA/CONFERENCE/

ТЕРНОПІЛЬ

2026

Почесні співголови конференції:

Десятнюк О.М., ректор Західноукраїнського національного університету, д-р економічних наук, професор;

Дивак М.П., д-р технічних наук, професор, проректор з наукової роботи ЗУНУ;

Голова конференції:

Березький О.М., д-р технічних наук, професор, професор кафедри комп'ютерної інженерії Західноукраїнський національний університет;

Заступники голови конференції:

Мельник Г.М., к.т.н, доцент, Західноукраїнський національний університет;

Піцун О.Й., к.т.н. доцент, Західноукраїнський національний університет;

ПРОГРАМНИЙ КОМІТЕТ КОНФЕРЕНЦІЇ

Склад програмного комітету:

Семанюк В.З., д-р економічних наук, професор, начальник науково-дослідної частини Західноукраїнського національного університету

Антощук С.Г., д.т.н, професор, Національний університет «Одеська політехніка»;

Баловсяк С.В., д.т.н, професор, Чернівецький національний університет

Бармак О.В., д.т.н., професор, Хмельницький національний університет

Батько Ю.М., к.т.н., доцент, Західноукраїнський національний університет

Березька К.М.,к.т.н.,доцент, Західноукраїнський національний університет

Винокурова О.А., д.т.н., професор, Львівський національний університет імені Івана Франка

Возна Н.Я., д.т.н., професор, Західноукраїнський національний університет;

Говорущенко Т.О., д.т.н., професор, Хмельницький національний університет;

Дубчак Л.О., к.т.н., доцент, Західноукраїнський національний університет;

Дунець Р.Б., д.т.н., професор, НУ "Львівська політехніка";

Ізонін І.В., д.т.н., доцент, НУ "Львівська політехніка";

Комар М.П., д.т.н, професор, Західноукраїнський національний університет;

Литвиненко В.І., д.н.т, професор, Національний технічний університет України "Київський політехнічний інститут";

Лупенко С.А., д.т.н., професор, Опольський технологічний університет, Польща;

Лящинський П.Б., доктор філософії з комп'ютерних наук, НУ "Львівська політехніка" ;

Мельникова Н.І., д.т.н., професор, НУ "Львівська політехніка" ;

Мулеса О.Ю., д.т.н., професор, Пряшівський університет, Словаччина

Пелешко Д.Д., д.т.н., професор, Львівський національний університет імені Івана Франка;

Сельський П.Р., д.м.н., професор, Тернопільський національний медичний університет імені І. Я. Горбачевського ;
Субботін С.О., д.т.н., професор, Національний університет «Запорізька політехніка» ;
Теслюк В.М., д.т.н., професор, НУ "Львівська політехніка" ;
Тимченко Л.І., д.т.н., професор, Державний університет інфраструктури та технологій;
Цмоць І.Г., д.т.н., професор, НУ "Львівська політехніка" ;
Якименко І.З., к.т.н., доцент, Західноукраїнський національний університет;
Яровий А.А., д.т.н., професор, Вінницький національний технічний університет ;
Яцків В.В., д.т.н., професор, Західноукраїнський національний університет.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ КОНФЕРЕНЦІЇ

Склад організаційного комітету:

Мельник Г.М. к.т.н., доцент, Західноукраїнський національний університет
Піцун О.Й. к.т.н., доцент, Західноукраїнський національний університет

Технічна підтримка:

Зінькевич О. В. студентка, Західноукраїнський національний університет
Кіт М. О. студентка, Західноукраїнський національний університет
Лебединський Т.В. студент,Західноукраїнський національний університет;
Кришталь Е.Р., студент, Західноукраїнський національний університет;

Метою конференції є представлення та обговорення наукових і практичних результатів, сприяння активізації творчої і інноваційної діяльності студентів, аспірантів і молодих вчених.
Конференція проводиться із залученням Ради Молодих Вчених та Студентського Наукового Товариства ФКІТ ЗУНУ.

IV Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ весна 2026), м. Тернопіль, ЗУНУ, 29 травня 2026 р. Тернопіль, 2026

Адреса:
Вул. О. Теліги, 8, корпус №6, Тернопіль
URL: https://ki.wunu.edu.ua/conference/
e-mail: kistudconference@gmail.com

Тези доповідей подаються за оригіналом рукопису

ЗМІСТ

Білак Ю.Ю.
к.ф.-м.н., доцент, кафедра програмного забезпечення систем, ДВНЗ "Ужгородський національний університет"

МЕТОДОЛОГІЯ СПЕКТРАЛЬНОГО АНАЛІЗУ ТА ПАРАМЕТРИЧНОГО СИНТЕЗУ ІЗ ВИКОРИСТАННЯМ ГІБРИДНИХ МОДЕЛЕЙ

Вступ. Сучасні дослідження у сфері оптичних багатошарових структур демонструють зростаючий інтерес до задач спектрального аналізу та параметричного синтезу тонких плівок [1]. Такі структури широко застосовуються в фотоніці, сенсорних системах, оптичних покриттях та нанотехнологіях, де критичним є точне керування спектральними характеристиками.

Аналіз сучасних публікацій показує, що традиційні чисельні методи, такі як метод матриць передачі (TMM), метод скінченних елементів (FEM), строгий хвильовий аналіз (RCWA) та метод скінченних різниць у часовій області (FDTD), забезпечують високу фізичну точність, однак характеризуються значними обчислювальними витратами, особливо при розв'язанні обернених задач [2,3].

З іншого боку, сучасні нейромережеві підходи (CNN, LSTM, автоенкодери) демонструють високу швидкодію та здатність до апроксимації складних залежностей, проте часто не гарантують фізичної коректності результатів [4].

Таким чином, актуальною є задача побудови гібридних методологій, що поєднують фізично обґрунтовані моделі та методи машинного навчання для підвищення точності, стійкості та ефективності розв'язання задач спектрального аналізу.

Постановка задачі. Об'єкт дослідження – багатошарові тонкоплівкові оптичні структури. Предмет дослідження – методи спектрального аналізу та параметричного синтезу з використанням фізичних і нейромережевих моделей. Мета дослідження – розробка методології, що забезпечує інтеграцію фізично обґрунтованих чисельних моделей та нейромережевих алгоритмів для ефективного розв'язання прямих, обернених задач спектроскопії та задач параметричного синтезу багатошарових тонких плівок.

Задача полягає у побудові єдиної інформаційної моделі, яка дозволяє:

- виконувати прямий спектральний аналіз;
- відновлювати параметри структури за спектральними даними;
- здійснювати параметричний синтез із заданими спектральними властивостями;
- забезпечувати обчислювальну ефективність та стійкість до шумів [3,5].

Основний матеріал. Запропонована методологія базується на формалізованому представленні у вигляді кортежу:

$$\Phi=\langle P,\ S(\lambda),\ Md,\ N,\ A,\ J,\ R\rangle$$

де:

- P – параметричний простір багатошарової структури (товщини шарів, показники заломлення, коефіцієнти поглинання);
- $S(\lambda)$ – спектральний простір (відбиття, пропускання, поглинання, еліпсометричні характеристики);
- Md – множина фізичних моделей (TMM, FEM, RCWA, FDTD, гібридні моделі);
- N – нейромережеві моделі (CNN, LSTM, CNN+LSTM, автоенкодери);
- A – алгоритми оберненого відновлення, чутливісного аналізу та оптимізації;
- $J(P)$ – функціонал нев'язки між експериментальними та розрахунковими спектрами;
- R – оператор відображення між параметричним і спектральним просторами.

Функціонал нев'язки визначається як:

$$J(P)=\|S_{\text{exp}}(\lambda)-S_{\text{calc}}(P,\ \lambda)\|^2+\Omega(P);$$

Методологія реалізує три основні класи задач:

1. Прямий спектральний аналіз. Прямий аналіз полягає у визначенні спектральних характеристик за заданими параметрами структури. Для цього використовуються фізично обґрунтовані моделі (TMM як базова, а також FEM, RCWA, FDTD). Гібридні підходи дозволяють враховувати складні геометричні та матеріальні неоднорідності.

2. Обернені задачі спектроскопії. Обернена задача полягає у відновленні параметрів структури за відомим спектром. Вона є некоректною та чутливою до шумів, тому потребує регуляризації та використання апріорної інформації. У запропонованій методології застосовуються нейромережеві моделі як початкове наближення з подальшим фізичним уточненням.

3. Параметричний синтез. Параметричний синтез передбачає визначення оптимальних параметрів структури для досягнення заданого спектрального відгуку. Це задача багатовимірної оптимізації з великою кількістю локальних мінімумів, яка вирішується із застосуванням адаптивних алгоритмів оптимізації та surrogate-моделей.

Особливістю підходу є *двоконтурна схема обчислень:*

- перший контур – нейромережеве швидке наближення параметрів;
- другий контур – фізично обґрунтоване уточнення шляхом мінімізації $J(P)$.

Такий метод дозволяє поєднати швидкість нейромережевих моделей із фізичною точністю чисельних методів.

Висновки. У роботі запропоновано гібридну методологію спектрального аналізу та параметричного синтезу багатошарових тонких плівок, що поєднує фізичні моделі та нейромережеві алгоритми в єдиній обчислювальній системі.

Основні наукові результати полягають у:

– формалізації єдиної моделі спектрального аналізу у вигляді кортежу Ф;

– розробці двоконтурної схеми обчислень;

– інтеграції чисельних та нейромережевих моделей у задачах оберненого відновлення;

– підвищенні точності та стійкості розв’язання задач спектроскопії;

– зменшенні обчислювальної складності за рахунок surrogate-моделей.

Практична цінність полягає у можливості застосування розробленої методології для аналізу експериментальних спектрів, відновлення параметрів реальних структур та проєктування оптичних багатошарових систем із заданими характеристиками.

Список літератури

1. Ma, T., Ma, M., & Guo, L. J. (2025). *Optical multilayer thin film structure inverse design: From optimization to deep learning*. iScience, 28(4), 112222. https://doi.org/10.1016/j.isci.2025.112222
2. Han, J. H. (2024). *Efficient inverse design of optical multilayer nano-thin films using neural network principles: Backpropagation and gradient descent*. Nanoscale, 16, 17165–17178. https://doi.org/10.1039/D4NR01667J
3. *Hybrid inverse design of photonic structures by combining optimization methods with neural networks*. (2022). Photonics and Nanostructures – Fundamentals and Applications, 52, 101073. https://doi.org/10.1016/j.photonics.2022.101073
4. Lininger, A., Hinczewski, M., & Strangi, G. (2021). *General inverse design of layered thin-film materials with convolutional neural networks*. ACS Photonics, 8(12), 3641–3650. https://doi.org/10.1021/acsphotonics.1c01498
5. Luce, A., et al. (2023). *Investigation of inverse design of multilayer thin-films with conditional invertible neural networks*. Machine Learning: Science and Technology, 4(1). https://doi.org/10.1088/2632-2153/acb48d

Чабан О.Р.
аспірант ФІТ ХНУ
Науковий керівник д.т.н., професор Манзюк Е.А., кафедра КН ХНУ

АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНТЕГРАЦІЇ ЗНАНЬ ПРОГРАМНОГО КОМПЛЕКСУ «IDK MEDICAL AI»

Актуальність. Медичні діагностичні комплекси переходять від ізольованого аналізу зображень до мультимодального опрацювання DICOM/NIfTI-даних, клінічних текстів, онтологій, процедурних правил і результатів попередніх обстежень. Класичні нейромережеві моделі забезпечують високу апроксимаційну здатність [1], але без явного введення знань залишаються чутливими до доменного зсуву, дефіциту розмічених вибірок і топологічних помилок сегментації [2, 3].

Публікації 2024–2025 рр. підтверджують загальну тенденцію: знання-орієнтовані графи, онтологічні вбудовування та гібридні моделі підвищують якість медичного логічного виведення, особливо за обмеженої кількості навчальних даних [4, 5]. Отже, актуальною є не окрема модель класифікації, а цілісна архітектура, яка керує потоком даних, формалізує експертні знання та забезпечує прозорий шлях від медичного сигналу до клінічного висновку.

Постановка задачі. Об'єкт дослідження – процеси інтеграції знань та опрацювання гетерогенних медичних даних у системах штучного інтелекту. Предмет дослідження – архітектурні засоби, математичні моделі та нейромережеві компоненти введення експертних знань у задачі сегментації, класифікації та аналізу клінічних текстів. Мета роботи полягає в обґрунтуванні архітектури «IDK Medical AI», яка поєднує попереднє опрацювання, хаб знань, дистиляцію, топологічно коректну сегментацію, графове логічне виведення та NLI-модуль для формування клінічно узгоджених результатів.

Основний матеріал. Інтелектуальна інформаційна система розглядається як конвеєр інтеграції знань, а не як набір автономних моделей. Її базова функція прийняття рішення подана у вигляді параметризованого відображення:

$$D = f_{\text{CAD}}(I; \theta, \mathscr{R}_{domain}). \quad (1)$$

У формулі (1) I позначає вхідне медичне зображення або мультимодальний тензор, θ – параметри моделі, а R_domain – формалізовані правила предметної області. На відміну від схеми D=f(I;θ), така постановка вводить медичні знання як рівноправний аргумент діагностичного рішення.

Проєктне рішення ґрунтується на трьох принципах: відокремлення знань від моделей, стандартизація проміжних артефактів та використання інтерпретованих виходів. Завдяки цьому один і той самий хаб знань може використовуватися в різних задачах: для дистиляції, формування SDF-карт, побудови графа пацієнта та перевірки логічної узгодженості клінічних текстів.

Рисунок 1 деталізує п'ять рівнів системи. Перший рівень формує потік багатовимірних радіологічних зображень і неструктурованих епікризів. Другий виконує стандартизацію форматів, нормалізацію інтенсивностей і токенізацію. Третій рівень є хабом знань: він генерує карти знакових відстаней, підтримує ансамбль моделей-вчителів та транслює медичні онтології у графові представлення. Четвертий рівень містить чотири сервіси: EMTKD, SKIF-Seg, KI-GCN і NLI. П'ятий рівень формує результати, які можна перевірити лікарем: компактну модель-учня, топологічно валідні маски, ймовірності патологічних станів і семантично узгоджені висновки.

Попереднє опрацювання зменшує варіативність обладнання й підготовлює дані до спільного використання в модулях сегментації, класифікації та NLP. Його можна подати так:

$$x' = \mathcal{T}_{aug}(\mathcal{T}_{norm}(x; \alpha_{norm}); \alpha_{aug}). \quad (2)$$

У формулі (2) $\mathcal{T}_{norm}$ нормалізує інтенсивності та просторове подання, а $\mathcal{T}_{aug}$ моделює допустимі збурення. Формула описує контрольовану трансформацію, яка зберігає клінічно

значущу структуру даних. Саме ця стадія зменшує відмінності між апаратами й готує дані до подальшої інтеграції знань.

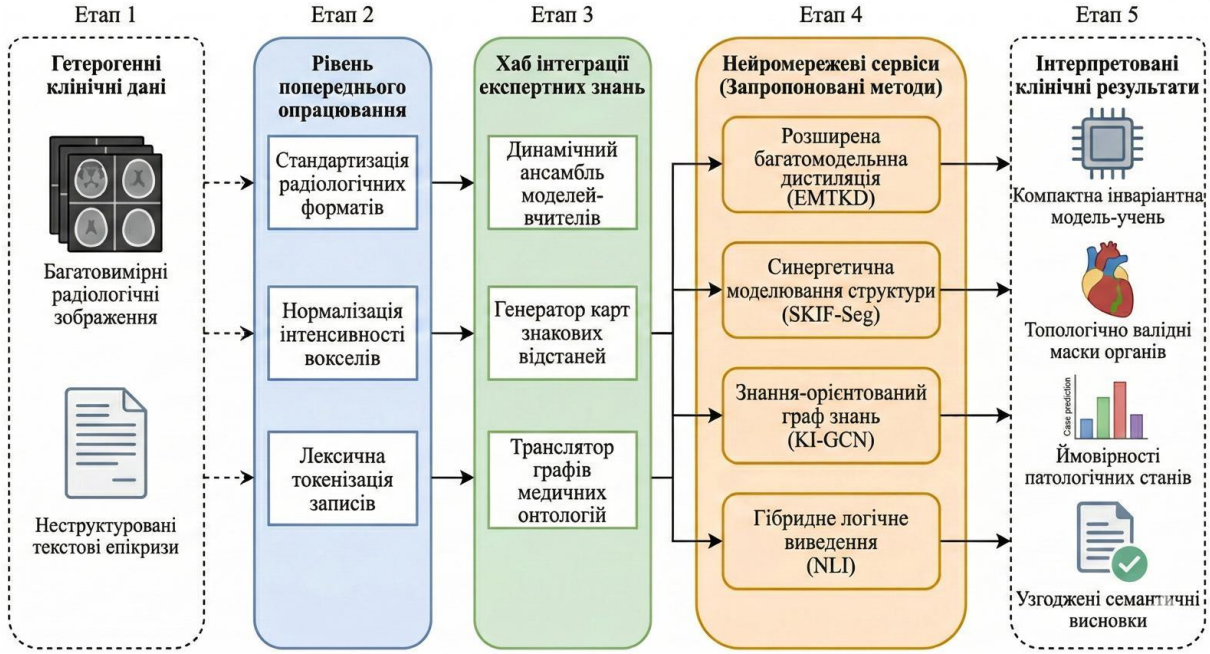


Рисунок 1 – Блок-схема архітектури інтелектуальної інформаційної системи інтеграції знань програмного комплексу «IDK Medical AI»

Модуль EMTKD призначений для перенесення знань від ансамблю моделей-вчителів до компактної моделі-учня. Для кожного прикладу агрегований розподіл м'яких міток формується динамічно:

$$p_T(y|x) = \sum_{m=1}^{M} w_m(x) p_m(y|x), \quad \sum_{m=1}^{M} w_m(x) = 1. \tag{3}$$

Ваги $w_m(x)$ відображають локальну надійність учителя для конкретного домену. Завдяки цьому система не усереднює несумісні прогнози, а пригнічує внесок моделей, що погано узгоджуються з поточним джерелом даних. Навчальна ціль компактної моделі подана як сума класифікаційної, дистиляційної та адаптаційної складових:

$$\mathscr{L}_{EMTKD} = \mathscr{L}_{CE}(y, p_S) + \lambda_{KD} T^2 KL(p_T^T \| p_S^T) + \lambda_{DA} \mathscr{L}_{adv}. \tag{4}$$

Перша складова контролює відповідність істинним міткам, друга передає узагальнені знання ансамблю, третя вирівнює доменні розподіли ознак. Ця комбінація пояснює приріст AUC-ROC за малої кількості розмічених зразків: модель-учень отримує не лише клас, а й структуру невизначеності ансамблю.

Модуль SKIF-Seg відповідає за анатомічно коректні маски. Його цільова функція поєднує піксельну точність і топологічні обмеження:

$$\mathscr{L}_{SKIF} = \mathscr{L}_{Dice} + \lambda_{CE} \mathscr{L}_{CE} + \lambda_T \mathscr{L}_{TAAC}, \tag{5}$$

$$\mathscr{L}_{TAAC} = \sum_{(a,b)\in\Omega}^{\Sigma_{ab}} max. \tag{6}$$

У формулах (5)–(6) $\mathscr{L}_{Dice}$ підвищує площу перекриття, $\mathscr{L}_{CE}$ стабілізує піксельну класифікацію, а $\mathscr{L}_{TAAC}$ штрафує порушення анатомічних відношень. Величини φ_a та φ_b є знаковими відстанями до меж структур, Ω задає пари структур із відомими обмеженнями, δ – допустимий топологічний зазор. Тому модель навчається не тільки "де знаходиться орган", а й "як орган має співвідноситися з іншими структурами".

Рисунок 2 показує перехід від МРТ-зображення та масок SKIF-Seg до графа пацієнта. Вузли графа відповідають анатомічним структурам серця, а їх ознаки поєднують морфологічні параметри та глибинні дескриптори. Ребра мають дві природи: просторову суміжність і клінічні кореляції, отримані з доменних знань. Така структура переводить класифікацію патологій із режиму «чорної скриньки» у режим реляційного міркування.

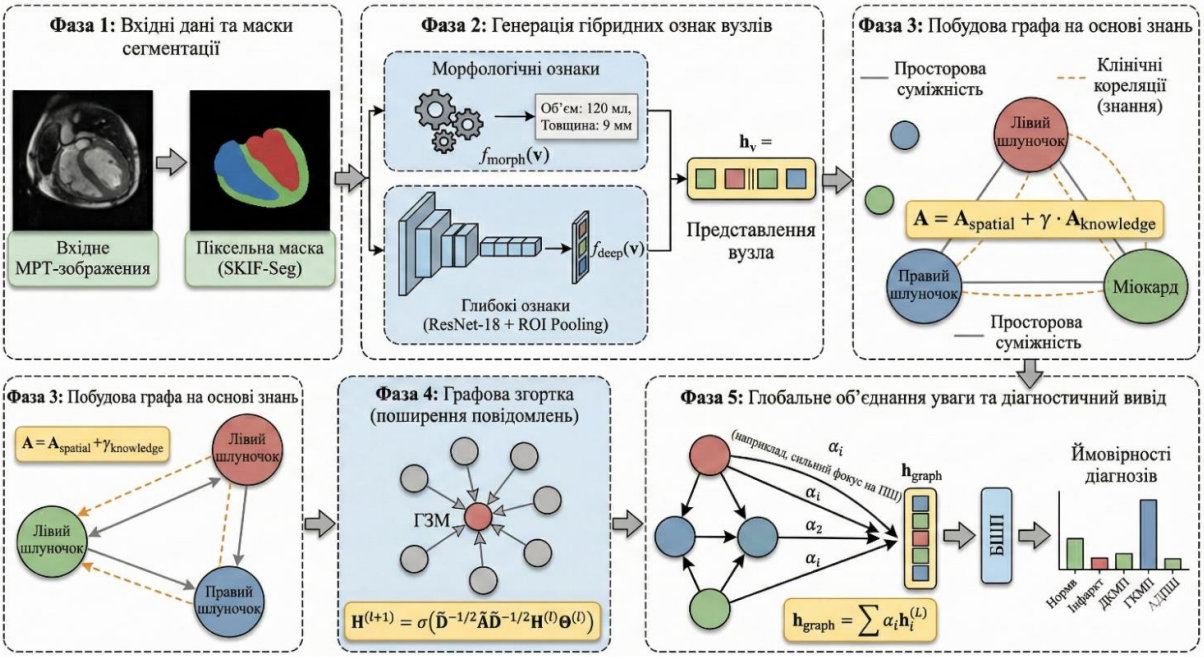


Рисунок 2 – Блок-схема графового логічного виведення модуля KI-GCN у системі «IDK Medical AI»

Таблиця 1 описує практичну придатність архітектури до локального інференсу. CUDA забезпечує найменшу медіанну затримку – 0,8 с, що у 6,6 раза швидше за CPU. DirectML має більшу затримку, ніж CUDA, але зберігає кросплатформену придатність для робочих станцій без NVIDIA GPU. Зростання пам'яті на GPU є прийнятним, оскільки інференс залишається в межах 4,1 ГБ, тобто система може працювати на типовому клінічному комп'ютері з дискретним прискорювачем.

Таблиця 1 – Пропускна здатність інференсу та використання ресурсів

Провайдер виконання	Медіана, с ↓	P95, с ↓	Пам'ять, ГБ ↓
CPU	5,3	6,6	3,2
CUDA	0,8	1,1	4,1
DirectML	1,2	1,6	3,8

Таблиця 2 поєднує ключові результати всіх функціональних блоків. Для EMTKD найбільший ефект спостерігається за 500 розмічених зразків: саме тут доменна адаптація та динамічне зважування вчителів мають максимальну цінність. Різниця між 81,45% і базовим діапазоном 72,15–77,10% показує, що архітектура не лише переносить знання, а й зменшує залежність від великого локального датасету.

Обмеження системи також мають архітектурний характер. Якість KI-GCN залежить від попередньої сегментації; помилки SKIF-Seg можуть переноситися в граф пацієнта. EMTKD потребує початкового ансамблю вчителів, що збільшує час навчання. OPNI залежить від повноти медичних онтологій і коректності вилучення понять. Тому практичне розгортання «IDK Medical AI» доцільно виконувати з журналюванням версій моделей, контролем домену даних і перевіркою калібрування ймовірностей.

Таблиця 2 – Узагальнені експериментальні результати модулів «IDK Medical AI»

Модуль	Набір / метрика	Базовий рівень	Результат	Інтерпретація
EMTKD	MIMIC-CXR, AUC-ROC, 500 зразків	72,15–77,10%	81,45±0,65%	Стійкість до дефіциту розмітки та доменного зсуву
OPNI	MedNLI, Accuracy / F1	79,73% / 78,23%	81,14% / 79,85%	Краще виявлення заперечень і семантичних суперечностей
SKIF-Seg	ACDC, ЛШ DSC / Mio HD95	93,0% / 9,8 мм	95,5% / 6,5 мм	Менше топологічних артефактів і точніші межі міокарда
KI-GCN	ACDC, Accuracy / AUC-ROC	85,0% / 0,92	94,0% / 0,97	Реляційне міркування підвищує точність і пояснюваність

Висновки. Суттєвою перевагою архітектури «IDK Medical AI» є наскрізне введення експертних знань у всі критичні етапи діагностичного конвеєра: нормалізацію даних, дистиляцію, сегментацію, графову класифікацію та семантичне виведення. Наукова новизна полягає в архітектурній інтеграції чотирьох знання-орієнтованих модулів, де кожен усуває окреме обмеження класичного глибокого навчання: нестачу розмітки, слабку інтерпретованість, топологічні артефакти або логічну неузгодженість текстових висновків.

Практичне значення роботи полягає в тому, що система демонструє прийнятні параметри інференсу, підвищує якість ключових метрик і зберігає можливість клінічної перевірки результату. Отримані результати підтверджують доцільність використання знання-інтегрованих архітектур як базового підходу для медичних діагностичних комплексів, у яких точність має поєднуватися з анатомічною валідністю та пояснюваністю.

Список літератури

1. O. Chaban, E. Manziuk, P. Radiuk, E. Zaitseva, and O. Markevych, “Intelligent information system for knowledge integration into artificial intelligence models,” in *Proc. 2nd Int. Workshop Adv. Appl. Inf. Technol.: AI & DSS*, Khmelnytskyi, Ukraine, Zilina, Slovakia, Dec. 5, 2025. Aachen: CEUR-WS.org, 2026, pp. 145–161. Accessed: Feb. 7, 2026. [Online]. Available: https://ceur-ws.org/Vol-4163/paper13.pdf

2. O. Chaban, E. Manziuk, and P. Radiuk, “Method of adaptive knowledge distillation from multi-teacher to student deep learning models,” *J. Edge Comput.*, vol. 4, no. 2, pp. 1–20, Aug. 2025. Accessed: Aug. 17, 2025. [Online]. Available: https://doi.org/10.55056/jec.978

3. P. Radiuk and O. Barmak, “Web-based information technology for classifying and interpreting early pneumonia based on fine-tuned convolutional neural network,” *Comput. Syst. Inf. Technol.*, vol. 3, no. 1, pp. 12–18, May 2021. Accessed: Sep. 4, 2025. [Online]. Available: https://doi.org/10.31891/CSIT-2021-3-2

4. M. Tomar, A. Tiwari, and S. Saha, “Towards knowledge-infused automated disease diagnosis assistant,” *Sci. Rep.*, vol. 14, Art. no. 13442, Jun. 2024. Accessed: May 13, 2026. [Online]. Available: https://doi.org/10.1038/s41598-024-53042-y

5. Y. Gao, R. Li, E. Croxford, J. Caskey, B. W. Patterson, M. Churpek, T. Miller, D. Dligach, and M. Afshar, “Leveraging medical knowledge graphs into large language models for diagnosis prediction: Design and application study,” *JMIR AI*, vol. 4, Art. no. e58670, Feb. 2025. Accessed: May 13, 2026. [Online]. Available: https://doi.org/10.2196/58670

Дмитрів Ю.Б.
аспірант 3 року навчання ІКНІ СШІ НУ«Львівська політехніка»
Науковий керівник: проф. Матвійчук Я. М., д.т.н.

ОПТИМІЗАЦІЯ ДІЯЛЬНОСТІ DEVOPS-ІНЖЕНЕРА ІЗ ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Вступ. Сучасна розробка програмного забезпечення нерозривно пов`язана з практиками DevOps, що поєднують процеси розробки (Development) та операційного супроводу (Operations) з метою скорочення циклу виведення продукту на ринок і підвищення його якості. Діяльність DevOps-інженера охоплює широкий спектр завдань: налаштування конвеєрів CI/CD, управління контейнерними середовищами, моніторинг хмарної інфраструктури, автоматизацію розгортання, аналіз інцидентів та усунення вразливостей. Зростаюча складність розподілених систем і збільшення обсягів телеметричних даних ставлять перед інженерами нові виклики, що важко вирішити виключно традиційними методами.

Активний розвиток великих мовних моделей (LLM) і генеративного штучного інтелекту (ШІ) відкриває принципово нові можливості для автоматизації рутинних операцій і підтримки прийняття рішень у DevOps-практиках. Дослідження провідних компаній у галузі (GitHub, Google, HashiCorp) свідчать про суттєве скорочення часу на виконання типових завдань при інтеграції ШІ-інструментів у робочі процеси інженерів [1, 2]. Водночас систематизовані підходи до такої інтеграції залишаються недостатньо опрацьованими як у науковій літературі, так і в інженерній практиці.

Таким чином, розроблення комплексної методики оптимізації діяльності DevOps-інженера із застосуванням сучасних інструментів штучного інтелекту є актуальним науково-практичним завданням, що має як теоретичне, так і прикладне значення.

Постановка задачі. Об’єктом дослідження є процес виконання операційних завдань DevOps-інженером у середовищах хмарних обчислень і контейнерної оркестрації. Предметом дослідження є методи та інструменти штучного інтелекту, що застосовуються для автоматизації, прискорення й підвищення якості виконання завдань DevOps-інженера. Метою роботи є розроблення методики інтеграції ШІ-інструментів у типові робочі процеси DevOps-інженера, що дозволяє знизити трудомісткість рутинних операцій і підвищити якість прийнятих рішень.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. виокремити ключові категорії завдань DevOps-інженера, що піддаються автоматизації засобами ШІ;
2. проаналізувати доступні LLM-інструменти й агентні фреймворки з точки зору їх застосовності в DevOps-практиках [3];
3. розробити формальну модель взаємодії інженера з ШІ-асистентом у розрізі типових сценаріїв;
4. оцінити ефективність запропонованих підходів за метриками часу виконання та якості артефактів.

Основний матеріал. Аналіз діяльності DevOps-інженера дозволив виокремити чотири класи завдань, найбільш перспективних з точки зору ШІ-оптимізації:

(А) генерація та перевірка конфігураційних файлів (Dockerfile, Kubernetes manifests, Terraform HCL);

(Б) аналіз журналів і метрик при розслідуванні інцидентів;

(В) написання й налагодження скриптів автоматизації (Bash, Python, PowerShell);

(Г) формування документації та інструкцій.

Кожен із цих класів характеризується великим обсягом шаблонних дій і необхідністю опрацювання значних текстових масивів.

Для формалізації процесу взаємодії інженера з ШІ-асистентом вводиться модель:

$$M = < T, P, A, R, Q >$$

де T - множина типів завдань, P - простір контекстних підказок (промптів), A - множина доступних ШІ-агентів і інструментів, R - простір результатів, $Q: T \times P \times A \rightarrow R$ - функція якості відображення задачі на результат через вибір агента.

Метою оптимізації є максимізація функції якості Q за обмежень на час виконання τ і обчислювальні ресурси C: знайти оптимальну пару $(p^*, a^*) = argmax\, Q(t, p, a)$ за умов $t(a) <= t\_max$ та $C(a) <= C\_max$.

Запропонована методика охоплює три рівні застосування ШІ. На першому рівні (реактивне асистування) використовуються inline-підказки IDE (GitHub Copilot, JetBrains AI Assistant) для генерації фрагментів коду і конфігурацій безпосередньо в редакторі. На другому рівні (ситуаційний аналіз) застосовуються LLM-чат-інтерфейси (Claude, ChatGPT) для аналізу аномалій у журналах, побудови гіпотез про першопричину інциденту (root cause analysis) і формування планів відновлення. На третьому рівні (агентна автоматизація) задіюються автономні ШІ-агенти з доступом до зовнішніх інструментів (виклик API моніторингових систем, виконання команд kubectl, terraform plan) через фреймворки типу LangChain або AutoGen [4].

Для класу задач (А) - генерації конфігураційних файлів - запропоновано шаблон структурованого промпту, що включає: опис цільової інфраструктури, технологічний стек, вимоги безпеки (принцип мінімальних привілеїв) та обмеження ресурсів. Експериментальна перевірка на наборі з 50 завдань з відкритих репозиторіїв показала, що використання структурованих промптів зменшує кількість ітерацій корекції згенерованого артефакту в середньому на 38% порівняно з непідготовленими запитами.

Для класу задач (Б) - аналізу інцидентів - розроблено пайплайн, в якому ШІ-модель отримує витяг із журналів у форматі структурованого контексту разом із описом симптомів і виводить ранжований список гіпотез першопричини з оцінкою вірогідності та рекомендованими кроками перевірки. Інтеграція такого підходу зі системами PagerDuty / Opsgenie дозволяє скоротити середній час до локалізації інциденту (MTTD) на 25–40% у порівнянні з ручним аналізом [5].

Важливим аспектом є управління якістю ШІ-генерованих артефактів. Запропоновано дворівневу схему верифікації: статичний аналіз (linting, security scanning засобами Checkov / Trivy) і семантична перевірка людиною-експертом. Це дозволяє зберегти контроль над критичними аспектами безпеки та відповідності стандартам при суттєвому зниженні ручного навантаження.

Висновки.У роботі запропоновано методику оптимізації діяльності DevOps-інженера на основі тришарової моделі застосування штучного інтелекту, що охоплює рівні реактивної асистенції, ситуаційного аналізу та агентної автоматизації. Наукова новизна полягає у формалізації процесу взаємодії інженера з ШІ через модель $M = < T, P, A, R, Q >$ і визначенні критеріїв оптимального вибору промпту та агента залежно від типу завдання.

Практичне значення отриманих результатів полягає в тому, що розроблені шаблони структурованих промптів і запропонований pipeline аналізу інцидентів можуть бути безпосередньо впроваджені в інженерних командах без зміни базової інфраструктури. Показано, що системне застосування ШІ-інструментів дозволяє зменшити трудомісткість генерації конфігурацій на 38% і скоротити час локалізації інцидентів на 25–40%. Подальші дослідження доцільно спрямувати на розроблення механізмів адаптивного вибору ШІ-агента залежно від поточного стану системи і накопиченої інженером контекстної бази знань.

Список літератури

1. Kalliamvakou E. Research: quantifying GitHub Copilot's impact on developer productivity and happiness // GitHub Blog. - 2022. - URL: https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/
2. Mozannar H., Bansal M., Fourney A., Horvitz E. Reading between the lines: Modeling user behavior and costs in AI-assisted programming // Proceedings of the ACM CHI Conference on Human Factors in Computing Systems. - 2023. - P. 1–15. DOI: 10.1145/3544548.3580997

3. Chase H. LangChain: Building applications with LLMs through composability. - 2022. - URL: https://github.com/langchain-ai/langchain
4. Wu Q., Bansal G., Zhang J. et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation // arXiv preprint arXiv:2308.08155. - 2023. - URL: https://arxiv.org/abs/2308.08155
5. Ahmed I., Bagheri E., Mokhtari G. AIOps: Real-world challenges and research innovations // IEEE Internet Computing. - 2023. - Vol. 27, No. 1. - P. 54–62. DOI: 10.1109/MIC.2022.3215728

Дутко Р.Р. (1)
студент 4 курсу АСУ ІКНІ НУЛП
Науковий керівник: к.т.н., доцент Цимбал Ю.В., кафедра АСУ, НУЛП

ГЕНЕРАЦІЯ ТРАНСФОРМОВАНИХ ЗАВДАНЬ З ПРОГРАМУВАННЯ ДЛЯ ЗАПОБІГАННЯ АВТОМАТИЧНОМУ РОЗВ'ЯЗАННЮ

Вступ. Поява доступних великих мовних моделей змінила характер виконання навчальних робіт з програмування. Сьогодні студент може не лише знайти фрагмент прикладу, а й отримати цілісний варіант розв'язку за коротким текстовим описом. Для дисциплін, пов'язаних з об'єктно-орієнтованим програмуванням (ООП) це особливо відчутно, оскільки значна частина типових вправ має передбачувану структуру: потрібно визначити сутності предметної області, описати їх властивості та реалізувати поведінку у вигляді методів. Такий формат добре сприймається генеративними моделями, тому відповідь може бути створена швидше, ніж студент встигне самостійно проаналізувати задачу.

Використання AI в освіті не є суто негативним явищем. Мовні моделі можуть пояснювати помилки, пропонувати ідеї, допомагати з навчальними прикладами та зменшувати бар'єр входу для початківців. Проте без зміни підходів до формулювання завдань вони також створюють ризик формального виконання роботи без засвоєння матеріалу. У наукових працях останніх років підкреслюється, що LLM можуть генерувати вправи, приклади коду і пояснення, але такі результати потребують перевірки викладачем та методичного контролю [1]. Огляди досліджень у галузі програмістської освіти також вказують на необхідність оновлення способів оцінювання знань в умовах активного використання генеративних інструментів [2].

Актуальність теми полягає в потребі перейти від простого виявлення або заборони використання AI до проєктування таких завдань, які зберігають навчальну мету, але вимагають від студента уважного читання та адаптації рішення до конкретних умов. У цій роботі запропоновано підхід, за якого початкова задача з ООП проходить контрольовані трансформації. Вони можуть змінювати контекст, формат подання, назви сутностей, точний контракт методів, додаткові обмеження і критерії перевірки. Завдяки цьому завдання залишається зрозумілим для людини, але перестає бути простим шаблоном для автоматичного розв'язання.

Постановка задачі. Об'єктом дослідження є процес підготовки, модифікації та перевірки навчальних завдань з об'єктно-орієнтованого програмування. Предметом дослідження є програмні засоби і методи трансформації, які дозволяють формувати індивідуалізовані варіанти задач, оцінювати їх стійкість до відповідей AI-моделей та підтримувати подальшу перевірку студентських рішень. Мета роботи полягає у створенні локальної настільної системи, яка забезпечує повний цикл роботи із завданнями: генерацію, трансформацію, benchmark-тестування, аналіз розв'язків, збереження даних і експорт результатів.

Для досягнення мети потрібно було розв'язати такі завдання: визначити вразливості типових вправ з ООП до автоматичного генерування коду; сформувати функціональні та нефункціональні вимоги; спроєктувати архітектуру настільного застосунку; реалізувати модулі генерації, трансформації, оцінювання, перевірки, збереження й експорту; організувати взаємодію з локальним AI-сервером; задати критерії оцінювання відповідей моделі; провести експериментальну перевірку на початкових і трансформованих умовах.

Основний матеріал. Розроблена система реалізована як локальний настільний застосунок мовою Java. Такий вибір зумовлений потребою в автономній роботі, можливості

зберігати дані локально та зручності демонстрації без розгортання вебсерверів. Java має строгу типізацію, зрілу екосистему бібліотек і природно відповідає предметній області, оскільки сама система працює із завданнями з ООП. У роботі використано сучасну платформу JDK 25 [3]. Інтерфейс побудовано на JavaFX 21, що дає змогу створювати багатовіконну навігацію, таблиці, списки, графіки, форми введення та текстові області в межах одного десктопного середовища [4].

Архітектура застосунку, наведена на рис. 1, поділена на кілька рівнів. Рівень представлення містить екрани реєстрації, входу, генерації, трансформації, тестування ефективності, перевірки студентського коду, історії, експорту та керування шаблонами. Рівень прикладної логіки координує основні сценарії: побудову задачі, застосування трансформерів, звернення до AI-моделі, автоматичне оцінювання відповіді, підготовку звіту та експорт. Рівень збереження побудовано на репозиторіях, тому прикладна логіка не залежить від конкретного сховища. У поточній версії використовується SQLite через JDBC: у локальній базі зберігаються користувачі, шаблони, згенеровані й трансформовані задачі, журнали генерацій та результати перевірки студентських робіт. Після публікації або подальшого розгортання системи цей рівень можна замінити на хмарне сховище, наприклад Firebase/Firestore або Google Cloud SQL, без зміни основних сценаріїв роботи.

Рис. 1. Контекстна діаграма архітектури системи

Модуль генерації підтримує два режими. Перший режим використовує заздалегідь підготовлені шаблони, теми ООП і профілі складності. Його перевага полягає у передбачуваності результату: викладач може контролювати, які знання перевіряє завдання і наскільки складною є сформована умова. Другий режим використовує LM Studio як локальний сервер мовної моделі. Система формує україномовний запит, передає його через OpenAI-сумісний REST API та отримує текст задачі або варіанта. Локальна інтеграція є важливою, оскільки навчальні матеріали не потрібно передавати у зовнішні хмарні сервіси, а параметри запиту можна налаштовувати для експериментів [5].

Ключовою частиною розробки став модуль трансформації. Він містить низку активних методів, які можна комбінувати залежно від теми задачі та бажаного рівня захисту. У поточній версії виділено шість категорій, до яких входять основні методи та їхні варіації: контекстний захист (таблиця індивідуальних варіантів, прив'язка до студента), точний контракт (фіксація назв класів і сигнатур методів, контрольні ключі, анти-generic обмеження), захист форматом (skeleton-код, пропуски в коді, фрагменти з помилками для виправлення), зрозумілість і самоперевірка (контрольні запитання та критерії перевірки), лінгвістичний зсув (зміна стилю формулювання) та Unicode-обфускація (гомогліфи, невидимі символи або маркування тексту). Щоб різні трансформації працювали узгоджено, у системі використано інтерфейс TaskTransformer. Він задає однакову модель підключення методів і дає змогу застосовувати їх як до структурованої чернетки задачі, так і до фінального тексту умови. Завдяки цьому перетворення не обмежуються заміною окремих слів. Вони можуть впливати на вимоги, перевірюваний контракт, формат відповіді та додаткові умови. При цьому важливим

принципом є збереження дидактичної мети: трансформована задача не повинна ставати хаотичною або незрозумілою, інакше вона ускладнюватиме навчання, а не лише автоматичне розв'язання.

Для оцінювання ефективності трансформацій реалізовано benchmark-модуль. Користувач обирає завдання, система надсилає його до LM Studio з детермінованими параметрами та отримує відповідь AI. Далі відповідь оцінюється за критеріями K1–K5: відповідність очікуваним класам, наявність потрібних методів, повнота виконання вимог, дотримання обмежень і якість коду. На основі цих критеріїв формується підсумкова оцінка та обчислюється ASR – частка випадків, у яких AI успішно впорався із завданням. Для інтерпретації результатів також використовується DSR, що показує частку спрацювання захисту.

Експериментальне тестування показало, що звичайні умови з ООП справді є вразливими до автоматичного розв'язання. Для початкових задач мовні моделі досягли ASR = 100%, тобто змогли сформувати прийнятні відповіді для всіх перевірених умов. Після застосування трансформацій показник знизився: для однієї моделі ASR становив 43%, для іншої – 57%. Відповідно, DSR становив 57% і 43%. Найбільш корисними виявилися не косметичні зміни, а ті методи, які впливають на контракт, формат і конкретні перевірювані вимоги. Це означає, що надійність захисту залежить не від довжини або складності тексту, а від того, наскільки чітко задача прив'язана до унікальних умов і критеріїв перевірки.

Окремий модуль призначений для аналізу студентських рішень. Він дозволяє вибрати завдання, вставити код і сформувати звіт перевірки. Система аналізує наявність очікуваних класів і методів, entry point, контрольних рядків, ознак надто загальних рішень та використання глобальних або статичних сховищ. Такий підхід доповнює benchmark-тестування: перший етап оцінює стійкість завдання до AI, а другий допомагає викладачу працювати з фактичними студентськими відповідями.

Для практичного використання передбачено експорт результатів. DOCX-файли формуються за допомогою Apache POI, PDF – за допомогою OpenPDF, а додаткові PNG-зображення можуть створюватися для UML-діаграм або допоміжних матеріалів. Це дає змогу використовувати систему не лише як дослідницький прототип, а і як інструмент підготовки навчальних матеріалів: викладач може створити варіанти, перевірити їх на стійкість, переглянути історію і зберегти готові файли для розповсюдження студентам.

Висновки. У роботі запропоновано підхід до підготовки трансформованих завдань з об'єктно-орієнтованого програмування, який спрямований на зменшення ефективності прямого автоматичного розв'язання великими мовними моделями. Наукова новизна полягає у поєднанні параметризованої генерації, набору контрольованих трансформацій, локального benchmark-тестування та формального оцінювання AI-відповідей за критеріями K1–K5.

Практична цінність розробки полягає в тому, що вона підтримує повний цикл роботи викладача із завданням: від створення умови до її експорту й аналізу студентського коду. Система не усуває проблему використання AI повністю, але надає інструмент для керованого зменшення ризику шаблонного розв'язання. Отримані результати підтвердили, що трансформації, орієнтовані на формат, точний контракт і конкретні вимоги, помітно знижують успішність автоматичної відповіді. Подальший розвиток може включати глибший AST-аналіз, підтримку більшої кількості мов програмування, інтеграцію з LMS і автоматичне формування тестів для перевірки студентських програм.

Список літератури

1. Sarsa S., Denny P., Hellas A., Leinonen J. Automatic Generation of Programming Exercises and Code Explanations using Large Language Models. arXiv, 2022. Доступ: https://arxiv.org/abs/2206.11861

2. Zhu M., Xu L., Ericson B. A systematic review of research on large language models for computer programming education. arXiv, 2025. Доступ: https://arxiv.org/abs/2506.21818

3. Oracle. Java Platform, Standard Edition & Java Development Kit Specifications. Version 25. 2025. Доступ: https://docs.oracle.com/en/java/javase/25/docs/specs/index.html

4. OpenJFX. JavaFX Documentation and Getting Started Guide. 2024. Доступ: https://openjfx.io/openjfx-docs/

5. LM Studio. OpenAI Compatibility Endpoints. 2025. Доступ: https://lmstudio.ai/docs/app/api/endpoints/openai

Komarnytska M. V.[1], Zvarych V. I.[2]
[1]4th-year student at ACS LPNU, [2]assistant at ACS LPNU
Scientific supervisor Ph.D., assistant Zvarych V. I., ACS LPNU

AN INTELLIGENT PNEUMONIA DETECTION AND LOCALIZATION SYSTEM USING YOLO11

Introduction. Pneumonia remains a critical global health challenge, representing a significant cause of morbidity and mortality across all age groups [1]. The pathology requires rapid and accurate identification to initiate timely therapy. Traditionally, chest X-ray (CXR) imaging serves as the primary diagnostic tool [2]. However, manual interpretation is labor-intensive and prone to subjective bias. The rapid evolution of Artificial Intelligence (AI) has introduced transformative possibilities in medical imaging. Modern Deep Learning architectures, specifically Convolutional Neural Networks (CNNs), have demonstrated remarkable capabilities in identifying subtle patterns within radiographic data [3]. Among these, object detection models have emerged as a superior alternative to simple classification, providing both the presence of a condition and its precise anatomical localization. The integration of Computer Vision into clinical workflows addresses the urgent need for decision-support systems that enhance diagnostic throughput while maintaining high precision [4].

Problem Statement. The primary objective of this research is to develop an intelligent system for automated pneumonia detection on chest X-ray images using the state-of-the-art YOLO11 model.

Object of research: The process of automated medical image diagnostics for respiratory diseases.

Subject of research: Methods and algorithms for real-time object detection and localization within the YOLO11 architecture applied to chest radiographs.

Purpose of research: To increase the efficiency of pneumonia screening by providing precise localization of inflammatory infiltrates through an accessible web-based platform.

Main Material. The proposed system utilizes the YOLO11 architecture [5], which introduces optimized feature extraction and a refined task-aligned head. A critical component of the pipeline is the medical preprocessing stage. To enhance the diagnostic quality of radiographs, the Contrast Limited Adaptive Histogram Equalization (CLAHE) filter is applied [6]. This technique improves local contrast and enhances definitions of pulmonary structures, which is essential for identifying subtle opacities [4]. The images are processed at a high resolution of 1024x1024 pixels to preserve fine clinical details that are often lost at lower resolutions.

The training strategy incorporates anatomical logic by specifically managing data augmentation. Techniques like mosaic augmentation and vertical / horizontal flips are disabled to maintain the anatomical integrity of the chest (e.g., ensuring the heart remains on the left side). To maximize the model's sensitivity, Test Time Augmentation (TTA) is implemented during the inference phase, which significantly improves the Recall metric by analyzing multiple augmented versions of the input image simultaneously.

The system is implemented as a web platform with a React.js frontend and a FastAPI backend. The FastAPI server handles the preprocessing (CLAHE) and neural network execution, providing the user with visualized results including bounding boxes and confidence scores. This allows clinicians to immediately identify areas of concern.

Conclusions. The development of the intelligent pneumonia diagnosis system using YOLO11 demonstrates the potential of deep learning in medical imaging. The implementation of CLAHE and high-resolution processing, combined with the integration of cloud technologies for data storage, ensures high diagnostic sensitivity and accessibility. Future perspectives involve expanding datasets to include more diverse pathological cases and further scaling the existing cloud architecture to support real-time multi-institutional deployment across various healthcare facilities.

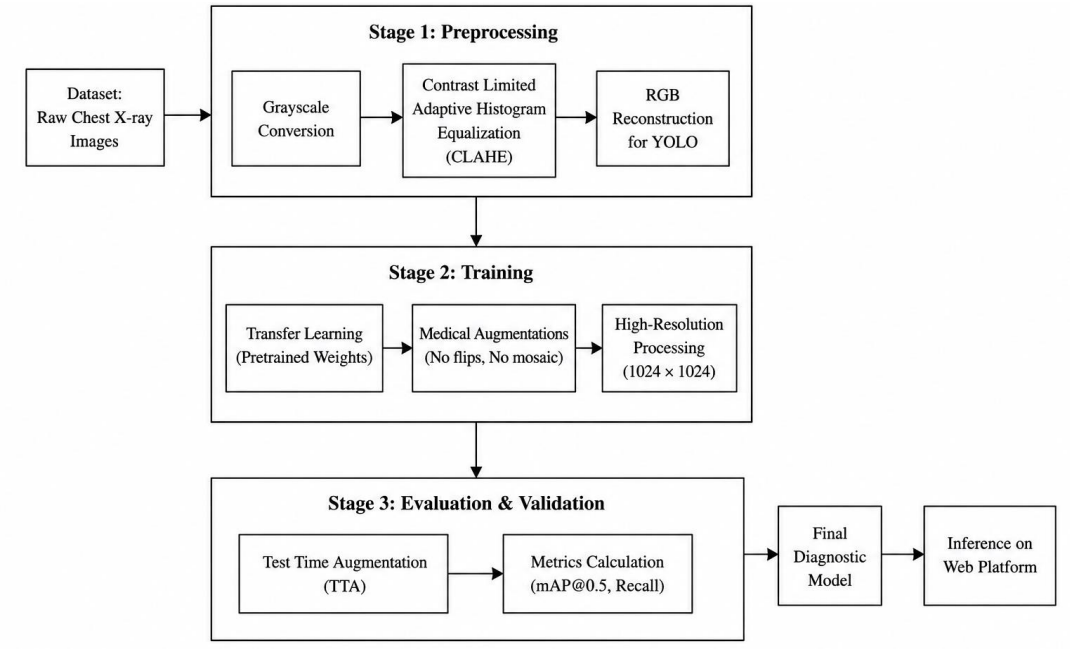


Figure 1 – AI-driven data processing pipeline for pneumonia detection

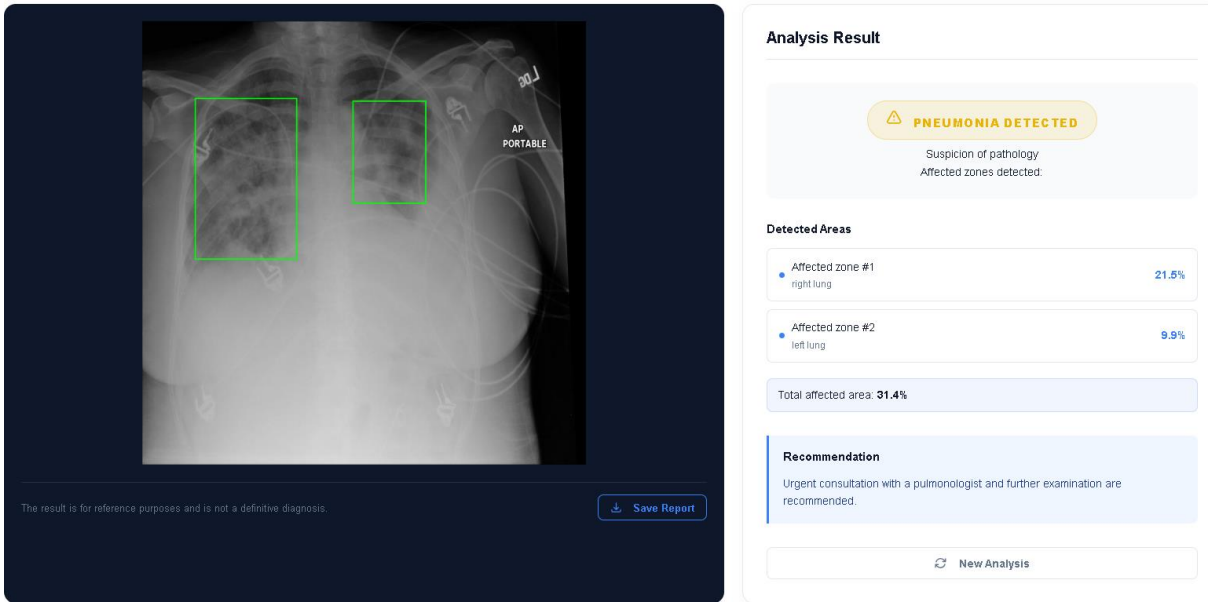


Figure 2 – Result of pneumonia detection on a chest X-ray image

References

1. Cillóniz, C., Fally, M., Evans, S. The pneumonia challenge: time to end the global neglect. *The Lancet Global Health*. **2025**. Vol. 13. e2000-e2001.

2. Slimi, H., Balti, A., Abid, S. Trustworthy pneumonia detection in chest X-ray imaging through attention-guided deep learning. *Scientific Reports*. **2025**. Vol. 15. 40029.

3. Xie, Y., Zhu, B., Jiang, Y., Zhao, B., & Yu, H. Diagnosis of pneumonia from chest X-ray images using YOLO deep learning. *Frontiers in neurorobotics*. **2025**. Vol. 19. 1576438.

4. Guan, L., Zhang, R., & Zhao, Y. YOLOv11-MFF: A multi-scale frequency-adaptive fusion network for enhanced CXR anomaly detection. *PloS one*. **2025**. Vol. 20. e0334283.

5. Ultralytics. YOLOv11: Real-time Object Detection and Beyond. Ultralytics Documentation. https://docs.ultralytics.com/models/yolo11.

6. Chen, R.-C., Dewi, C., Zhuang, Y.-C., Chen, J.-K. Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models. *IEEE Access*. **2023**. DOI: 10.1109/ACCESS.2023.3309410.

Мавко У. В.
студентка 4 курсу кафедри АСУ, ІКНІ, НУ «Львівська політехніка»
Науковий керівник: асистент Глод С. І., кафедра АСУ, НУ «Львівська політехніка»

ВЕБОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ ОСВІТНЬО-ВИХОВНОЮ ТА ОРГАНІЗАЦІЙНОЮ ДІЯЛЬНІСТЮ НСОУ «ПЛАСТ»

Вступ. Національна скаутська організація України «Пласт» – одна з найбільших молодіжних організацій країни, що охоплює понад 10 000 активних учасників щорічно. Центральним елементом пластової виховної методики (скаутської системи особистісного розвитку) є система проб – структурованих програм особистісного розвитку, що існують для кожного вікового рівня: пташацтва, новацтва та юнацтва. Кожна проба складається із завдань, виконання яких пластун фіксує, а виховник перевіряє й підтверджує. На практиці цей процес досі ведеться переважно вручну або через загальні інструменти, що не підтримують рольову модель виховника-пластуна і логіку підтвердження завдань, притаманну пластовій системі. Аналіз наявних програмних рішень виявив суттєві обмеження кожного з них.

Google Sheets дозволяє вести довільні таблиці, але не підтримує рольову модель, не має механізму підтвердження завдань і не масштабується для організацій із сотнями учасників через відсутність структурованої рольової моделі. *Moodle* – це потужна система управління навчанням (LMS), однак орієнтована на формальне навчання з лінійними курсами й не підтримує скаутську логіку «виконав пробу -> виховник підтвердив -> проба зарахована». *ScoutBook* розроблено для Boy Scouts of America і не відповідає структурі українського пластування – ані термінологічно, ані методично. Детальне порівняння наведено у табл. 1.

Таблиця 1. Порівняльний аналіз програмних рішень за функціональними критеріями

Критерій	Google Sheets	Moodle	ScoutBook (BSA)	Розроблена система
Облік пластових проб (новацтво/юнацтво)	✗	✗	частково	✓
Ролі зі специфічною взаємодією	✗	частково	✗	✓
Автоматичний розрахунок прогресу	✗	✓	✓	✓
Управління гуртками та подіями	✗	✗	✗	✓
Підтвердження завдань виховником	✗	частково	✓	✓
Навчальні матеріали до проб	✗	✓	частково	✓
Локалізація під НСОУ «Пласт»	✗	✗	✗	✓

Актуальність. Потреба у спеціалізованій системі є особливо гострою сьогодні з кількох причин. По-перше, впродовж 2022-2024 років Пласт суттєво розширив географію: до організації долучилися тисячі учасників з регіонів, де Пласт раніше був слабо представлений, і виховники стикнулися з різким зростанням кількості підопічних, яких потрібно відстежувати одночасно. По-друге, перехід до цифрового документообігу в освітніх організаціях є

глобальним трендом - дослідження підтверджують, що цифровізація управління в освітніх організаціях суттєво знижує адміністративне навантаження і підвищує залученість учасників [1], а онлайн-інструменти для молодіжних організацій зокрема демонструють значний потенціал для підтримки колаборації між учасниками та координаторами [2]. По-третє, технологічна зрілість сучасного стеку технологій – React/Next.js, хмарна PostgreSQL та JWT-автентифікація – дозволяє реалізувати таку систему без значних витрат на інфраструктуру, що підтверджується досвідом впровадження хмарних освітніх систем [4]. Жоден із доступних продуктів не враховує специфіку пластових проб, тому розробка власного рішення є єдиним, що повністю відповідає потребам НСОУ «Пласт».

Постановка задачі. Метою роботи є проєктування та реалізація веборієнтованої інформаційної системи для автоматизації управління освітньо-виховною та організаційною діяльністю НСОУ «Пласт». Система охоплює два вікові рівні: новацтво і юнацтво (тобто рівні, для яких повністю визначено структуру проб та завдань) , а також підтримує три ролі з чіткою специфікою взаємодії:

- Пластун – переглядає перелік своїх проб і завдань, фіксує виконання, відстежує прогрес у відсотках, отримує доступ до навчальних матеріалів для підготовки до здачі проби
- Виховник – бачить зведену таблицю прогресу всіх учасників свого гуртка, підтверджує або відхиляє виконані завдання з можливістю залишати текстовий відгук, керує розкладом подій і складом гуртка.
- Адміністратор – управляє структурою гуртків, редагує бібліотеку проб і завдань, додає навчальні матеріали, керує обліковими записами всіх користувачів системи.

Ключова технічна проблема – це реалізація узгодженої рольової моделі доступу, де кожна роль отримує чітко визначений набір даних і дій, що реалізовано через перевірку JWT-токенів на рівні middleware для кожного маршруту API.

Основна частина. Систему побудовано за клієнт-серверною архітектурою, що представлена на рис. 1. Фронтенд реалізовано на React та Next.js - фреймворку, що забезпечує серверний рендеринг (SSR), файлову маршрутизацію та оптимізоване завантаження ресурсів. Серверна частина побудована на Express (Node.js) і надає REST API (програмний інтерфейс передачі даних на основі архітектурного стилю REST) для всіх сутностей системи. Для взаємодії з базою даних використано Drizzle ORM над хмарною PostgreSQL (Neon), що забезпечує типобезпечні запити та автоматичне масштабування. Автентифікацію реалізовано через JWT-токени з перевіркою ролі на рівні проміжного програмного забезпечення (middleware) для кожного маршруту API [3]. Користувачі взаємодіють із веб-інтерфейсом через HTTPS, клієнт надсилає запити до API у форматі JSON, сервер виконує бізнес-логіку та вертається до PostgreSQL для збереження і отримання даних. Функціональне охоплення системи представлено на рис. 1.

Основна функціональність реалізується через сім API-модулів: автентифікація, користувачі, проби, завдання, вмілості, гуртки та події. Типовий сценарій взаємодії: пластун відкриває пробу, позначає завдання як виконане – воно переходить у стан «очікує підтвердження»; виховник переглядає виконання і підтверджує або відхиляє його з коментарем. Система автоматично перераховує відсоток виконання проби після кожного підтвердження. Статуси завдань зберігаються окремо від загального прогресу по пробах, що дозволяє виховнику одночасно відстежувати стан кожного учасника як у розрізі окремих завдань, так і цілих проб.

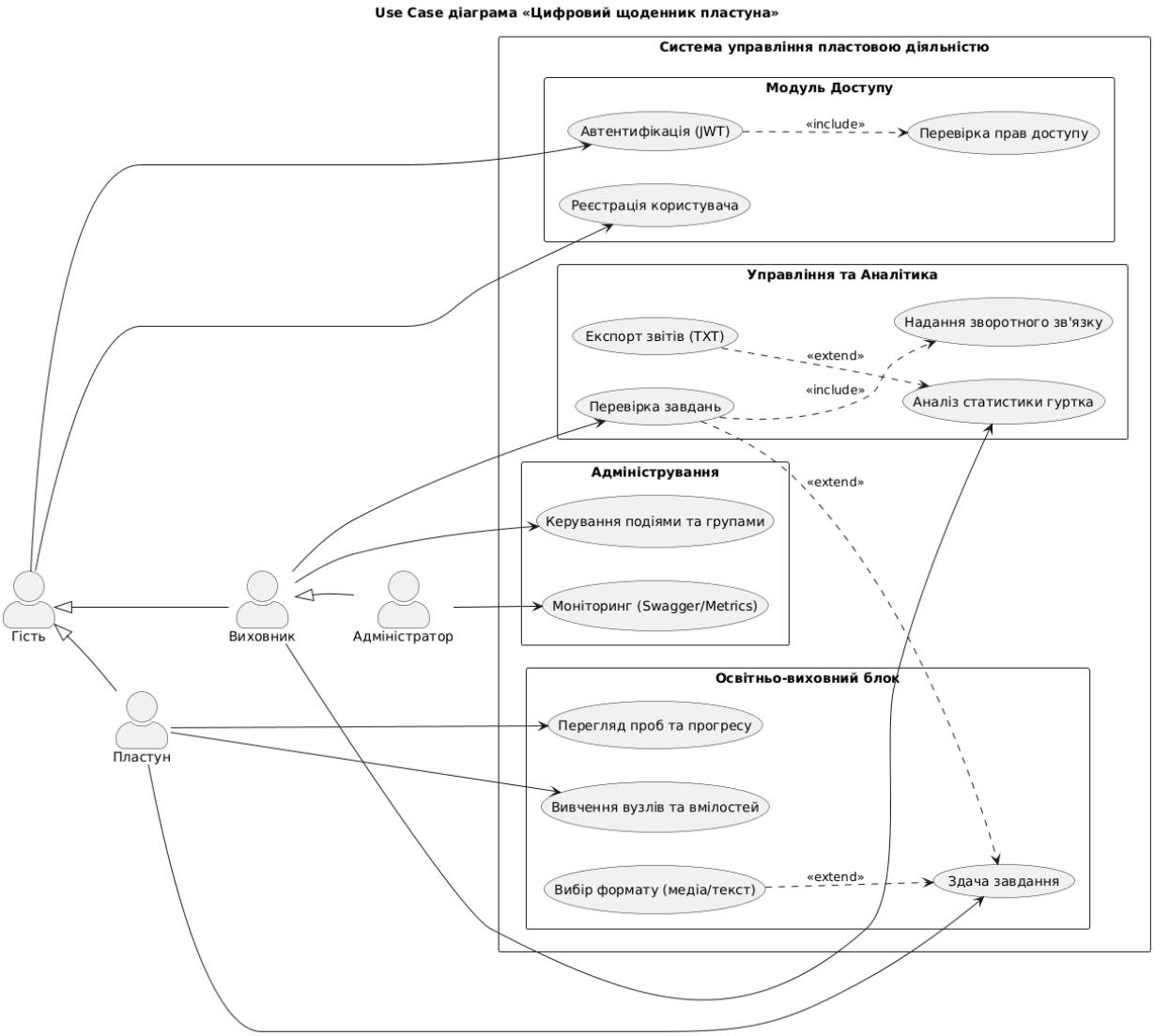


Рисунок 1. Use case діаграма системи «Цифровий щоденник пластуна»

Інтерфейс розроблено відповідно до бренд-буку НСОУ «Пласт» – з фірмовими кольорами, шрифтами та графічними елементами. Навігація побудована за рольовим принципом: пластун бачить лише власні проби та прогрес, виховник – зведену панель моніторингу гуртка. Приклад інтерфейсу наведено на рис. 2.

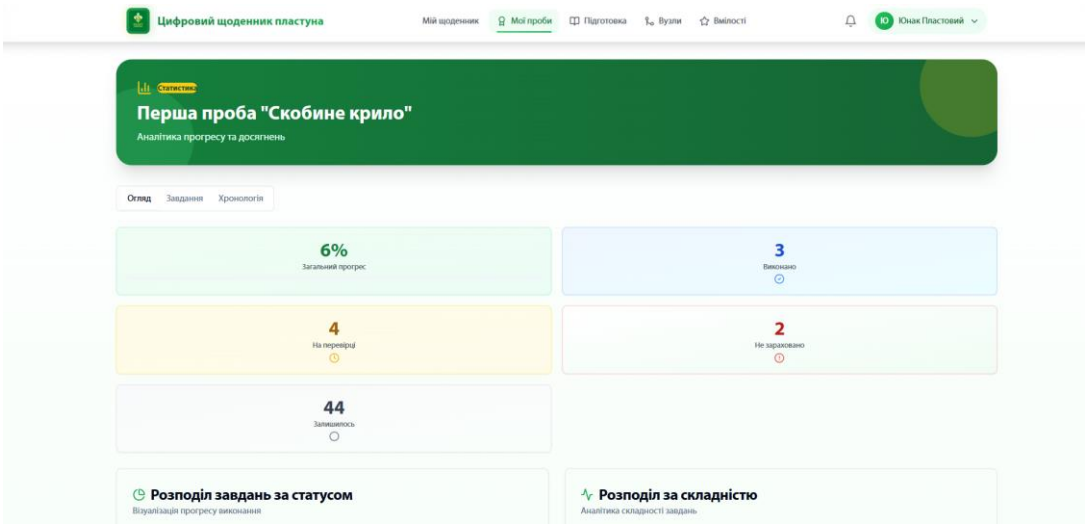


Рисунок 2. Зведена панель прогресу пластуна: завдання і статуси виконання

Висновки. У роботі розроблено веборієнтовану інформаційну систему управління освітньо-виховною та організаційною діяльністю НСОУ «Пласт». Порівняльний аналіз наявних рішень показав, що жодне з них не підтримує пластову логіку проб, рольову модель взаємодії та прив'язку до структури НСОУ, що обґрунтовує необхідність розробки спеціалізованої системи. Реалізована клієнт-серверна архітектура на основі Next.js, Express та хмарної PostgreSQL (Neon) із JWT-автентифікацією і рольовим розмежуванням доступу на рівні API забезпечує масштабованість для організацій із тисячами учасників. Система охоплює сім функціональних модулів повного циклу управління освітньо-виховною діяльністю для рівнів новацтва і юнацтва, а модуль відстеження прогресу із чотирьохстатусною моделлю завдань і автоматичним розрахунком відсотка виконання проби дозволяє виховнику контролювати стан усіх учасників гуртка через єдину панель моніторингу. Перспективою подальшого розвитку є інтеграція з хмарним файловим сховищем Cloudinary для підтримки завантаження навчальних матеріалів безпосередньо через інтерфейс системи.

Список літератури

1. Pacheco A., Yupanqui R., Mogrovejo D., Garay J., Uribe-Hernández Y. Impact of digitization on educational management: Results of the introduction of a learning management system in a traditional school context // Computers in Human Behavior Reports. - 2025. - Vol. 18. - Стаття 100592.
2. Cheng X., Fu S., de Vreede G. J. Is online collaboration process suitable for digital youth organization? A design approach // Journal of Electronic Business & Digital Economics. - 2022. - Т. 1, № 1-2. - С. 66-83.
3. Bucko A., Vishi K., Krasniqi B., Rexha B. Enhancing JWT Authentication and Authorization in Web Applications Based on User Behavior History // Computers. - 2023. - Vol. 12, № 4. - P. 1–15. DOI: 10.3390/computers12040078.
4. El Koshiry A., Eliwa E., Abd El-Hafeez T., Tony M. A. A. Effectiveness of a Cloud Learning Management System in Developing the Digital Transformation Skills of Blind Graduate Students // Societies. - 2024. - Т. 14, № 12. - Стаття 255

Слободянюк А.О.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

РОЗРОБКА ВЕБСАЙТУ ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ «ІКСМ»

Вступ. Цифровізація наукової діяльності є одним із ключових напрямів розвитку сучасної академічної спільноти. Організація наукових конференцій традиційно вимагає значних організаційних зусиль: обробка поданих матеріалів, комунікація з учасниками, публікація збірників та ведення архіву. Автоматизація цих процесів засобами веб-технологій дозволяє суттєво знизити трудовитрати організаторів та підвищити зручність для учасників. Актуальність роботи зумовлена відсутністю україномовного вебресурсу для конференції ІКСМ та необхідністю забезпечення зручного електронного прийому наукових матеріалів .

Постановка задачі. Об'єкт дослідження – процес організації та інформаційного забезпечення Всеукраїнської науково-практичної конференції «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ). Предмет дослідження – методи та засоби розробки вебсистем для автоматизації процесів подачі та публікації наукових матеріалів. Мета дослідження – розробити вебсайт конференції ІКСМ, що забезпечує зручний доступ до інформації про захід, електронний прийом матеріалів через форму подачі та публічний архів збірників.

Основний матеріал. Для визначення вимог до розроблюваного сайту проведено порівняльний аналіз існуючих платформ: порталу ДНТБ України [1], платформи International Scientific Unity та сервісу Confcontact. Аналіз показав, що всі розглянуті рішення орієнтовані на

англомовну аудиторію та міжнародні конференції великого масштабу, не підтримують повної кастомізації та є платними або мають обмежений функціонал для конференцій університетського рівня. Порівняльний аналіз наведено у таблиці 1.

Таблиця 1 – Порівняльний аналіз існуючих платформ

Критерій	Портал ДНТБ	International Scientific Unity	Confcontact	Розроблений сайт
Мова інтерфейсу	Українська	Англійська	Англійська	Українська
Подача матеріалів	Обмежена	Реалізована	Частково	Так (форма)
Архів публікацій	Є	Є	Обмежений	Є (PDF)
Авторизація	Є	Є	Є	Не потрібна
Кастомізація	Обмежена	Обмежена	Обмежена	Повна
Вартість	Безкоштовно	Платно	Платно	Власна розробка

Для розробки сайту обрано платформу WordPress 6.9.4 [2] у поєднанні з темою Astra [3] та конструктором сторінок Elementor [4]. Серверне середовище – XAMPP з веб-сервером Apache та СУБД MySQL. Для реалізації форми подачі наукових матеріалів використано плагін Forminator [5]. Усі компоненти є безкоштовними з відкритим вихідним кодом, що суттєво знизило вартість розробки .

Вебсайт побудований за клієнт-серверною архітектурою: браузер відвідувача надсилає HTTP-запит до веб-сервера Apache, PHP-інтерпретатор обробляє запит засобами WordPress, звертається до бази даних MySQL та повертає сформовану HTML-відповідь. Загальну архітектуру системи зображено на рисунку 1.

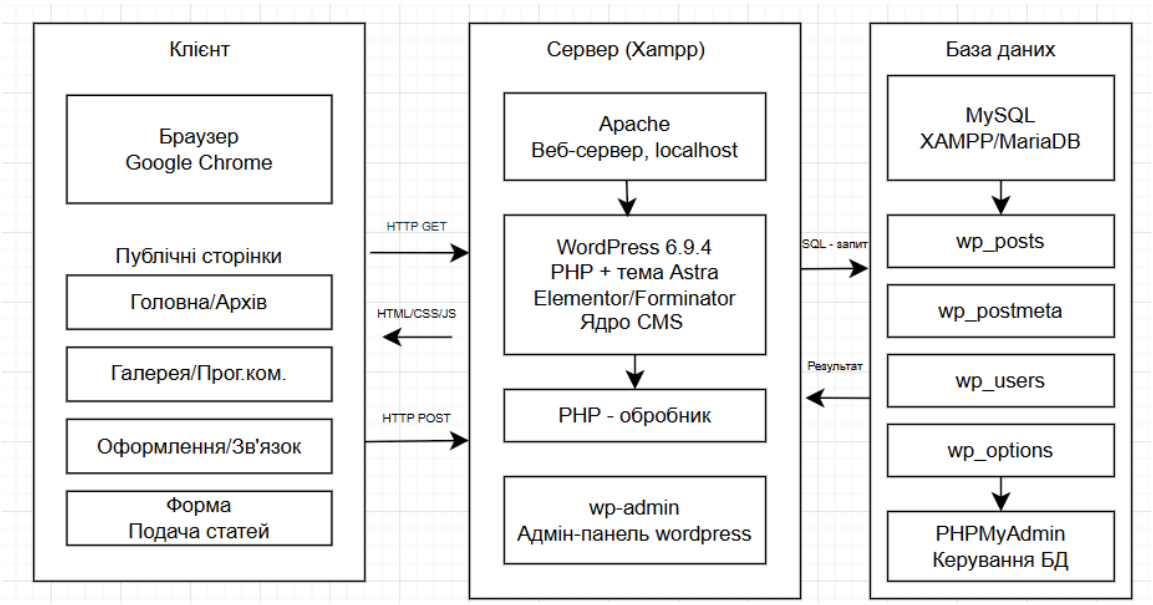


Рисунок 1 – Загальна архітектура клієнт-серверної системи вебсайту конференції

Публічна частина сайту складається з семи сторінок, доступних без авторизації. Адміністративна частина – стандартна панель WordPress, захищена автентифікацією. Структуру сайту наведено у таблиці 2.

Таблиця 2 – Структура вебсайту конференції ІКСМ

Сторінка	Функціональність	Доступність
Головна	Загальна інформація, секції, час і місце	Публічна
Архів	Збірники конференцій у форматі PDF	Публічна

Продовження Таблиці 2

Сторінка	Функціональність	Доступність
Галерея	Фотоматеріали з конференцій по роках	Публічна
Програмний комітет	Склад комітету з фото та регаліями	Публічна
Оформлення	Вимоги до статей, шаблон Word	Публічна
Зв'язок	Контакти, Google Maps, соцмережі	Публічна
Форма	Подача наукових матеріалів	Публічна
wp-admin	Керування вмістом та заявками	Лише адміністратор

Ключовою функціональною складовою сайту є форма подачі наукових матеріалів, реалізована засобами плагіну Forminator . Форма містить чотири поля: завантаження файлу (Word або PDF), ПІБ учасників, електронна адреса та поле «ВУЗ, Група, Науковий керівник». Вигляд форми на сторінці сайту наведено на рисунку 2.

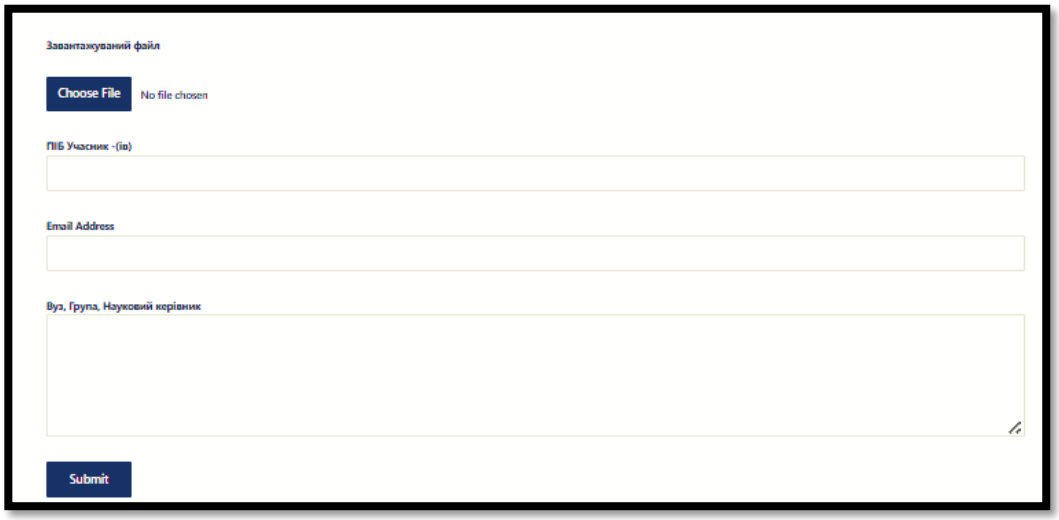


Рисунок 2 – Форма подачі наукових матеріалів

Валідація форми реалізована на двох рівнях: JavaScript перевіряє коректність полів на стороні браузера, PHP-обробник повторно перевіряє MIME-тип завантаженого файлу на сервері. Подані заявки зберігаються у базі даних WordPress та доступні адміністратору через розділ Submissions у панелі Forminator. Головну сторінку вебсайту наведено на рисунку 3.

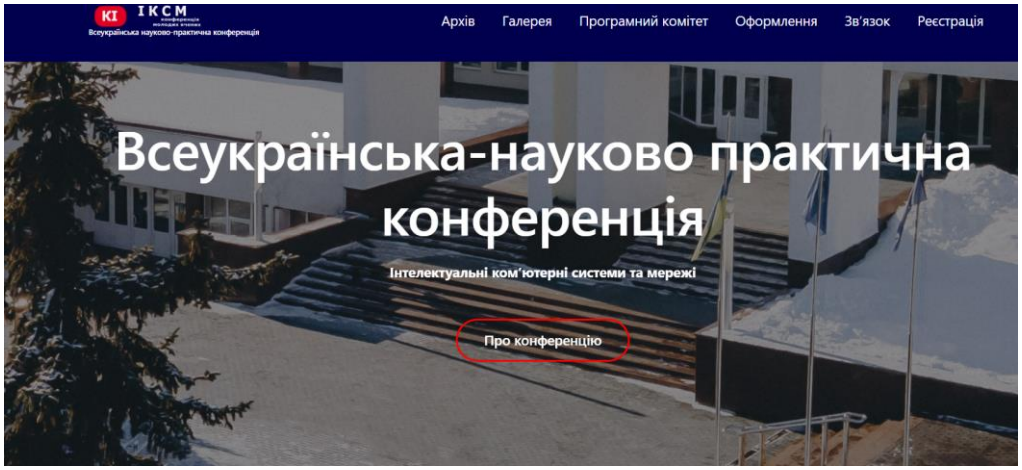


Рисунок 3 – Головна сторінка вебсайту конференції ІКСМ

Станом на травень 2025 року архів сайту містить матеріали десяти конференцій (з травня 2023 по листопад 2025), що підтверджує практичне впровадження розробленого ресурсу. Сторінку архіву зображено на рисунку 4.

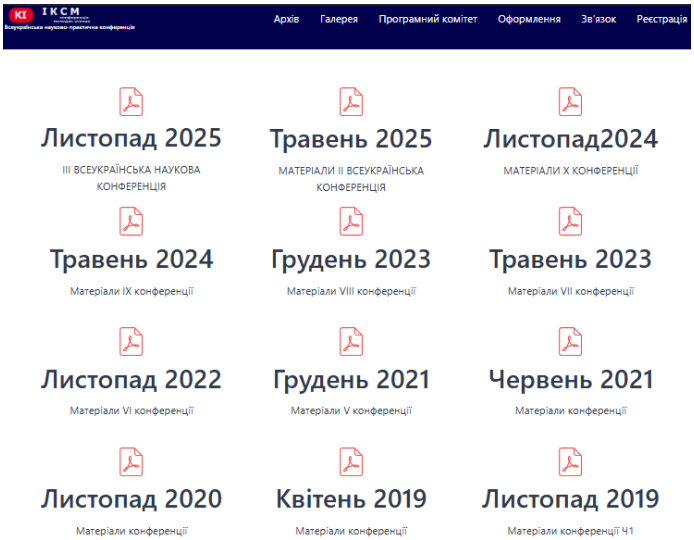


Рисунок 4 – Сторінка «Архів» з переліком збірників конференцій

Висновки. У роботі розроблено вебсайт Всеукраїнської науково-практичної конференції «Інтелектуальні комп'ютерні системи та мережі» на платформі WordPress із використанням теми Astra, конструктора Elementor та плагіну Forminator. Виконано порівняльний аналіз існуючих платформ, що підтвердив доцільність власної розробки. Реалізовано форму електронної подачі наукових матеріалів з дворівневою валідацією та збереженням заявок у базі даних. Використання виключно безкоштовного програмного забезпечення дозволило мінімізувати витрати на розробку. Сайт впроваджено в реальну діяльність конференції та активно використовується учасниками і організаторами з 2023 року.

Список літератури

1. WPMU DEV. Forminator Plugin Documentation. URL: https://wpmudev.com/docs/wpmu-dev-plugins/forminator (дата звернення: 16.05.2025).

2. Elementor Ltd. Elementor Knowledge Hub. URL: https://elementor.com/help (дата звернення: 16.05.2025).

3. WordPress Foundation. WordPress Documentation. URL: https://developer.wordpress.org (дата звернення: 16.05.2025).

4. Державна науково-технічна бібліотека України : веб-сайт. URL: https://dntb.gov.ua (дата звернення: 05.04.2025).

5. Brainstorm Force. Astra Theme Documentation. URL: https://wpastra.com/docs (дата звернення: 16.05.2025).

Сорочан Н.О.(1), Ніколенко В.В. (2)
1) студент 4 курсу кафедри ІУСТ ДВНЗ "Ужгородський національний університет", 2) к.ф.м.н. доцент кафедри ІУСТ ДВНЗ "Ужгородський національний університет"
Науковий керівник к.ф.м.н., доцент Ніколенко В.В., кафедра ІУСТ "Ужгородський національний університет"

ПОРІВНЯЛЬНИЙ АНАЛІЗ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТІВ УКРАЇНСЬКОЇ ДАКТИЛЬНОЇ АБЕТКИ

Вступ. За даними ВООЗ на 2026 рік у світі понад 400 мільйонів людей потребують реабілітації для лікування втрати слуху. Попри активний розвиток систем розпізнавання жестової мови у світі, українська дактильна абетка залишається мало представленою у відкритих датасетах та дослідженнях. Аналіз сучасних досліджень свідчить про стрімкий перехід від аналізу повних RGB-зображень до використання скелетних моделей рук, що дозволяє суттєво знизити обчислювальні витрати та підвищити інваріантність до умов освітлення і фону [1]. Зокрема, у роботі Samaan та ін. [2] продемонстровано, що поєднання координат ключових точок рук із рекурентними архітектурами (GRU, LSTM, BiLSTM) забезпечує точність понад 99% на динамічних жестах. Окрему увагу в літературі приділено задачам для малодосліджених мов: Gonfa та ін. [3] встановили, що skeleton-based підхід на основі MediaPipe дає 94% точності в межах одного диктора, проте лише 73% на нових користувачах, що підкреслює важливість різноманітності тренувального датасету. Uddin та ін. [4] підтвердили ефективність системи MediaPipe+LSTM для жестової мови малих мов з точністю 95%. Це свідчить про актуальність задачі та доцільність порівняльного аналізу архітектур саме для української дактильної абетки.

Постановка задачі. Об’єктом дослідження є процес розпізнавання жестів української дактильної абетки: системи ручних знаків, де кожна конфігурація кисті відповідає окремій літері українського алфавіту. Предметом дослідження є архітектури нейронних мереж LSTM, BiLSTM та 1D CNN у застосуванні до розпізнавання часових послідовностей ключових точок кисті. Метою роботи є порівняльний аналіз зазначених архітектур за критеріями точності класифікації та macro F1-score на власному датасеті відеопослідовностей жестів.

Основний матеріал. Для проведення порівняльного аналізу використано власний набір даних, що містить відеозаписи жестів 33 літер української дактильної абетки (23 статичних та 10 динамічних), по 65 відеопослідовностей на кожну літеру із залученням двох дикторів за умов різного освітлення: природного та штучного. Загальний обсяг датасету становить 2145 відеозаписів. Такий підхід до збору даних підвищує стійкість моделі до варіацій умов зйомки, що є важливим для практичного застосування. Для навчання моделей набір даних було розбито на train/val/test вибірки у співвідношенні 72/8/20 на рівні відеозаписів, що унеможливлює витік даних між вибірками.

Для вилучення ознак застосовано бібліотеку MediaPipe Hands, яка визначає 21 ключову точку кисті руки у тривимірному просторі. Вхідний вектор ознак формується як конкатенація x, y, z-координат однієї руки (63 значення) та вектора швидкості зміни координат між суміжними кадрами, що дає фінальний вектор розмірністю 126. Включення швидкості дозволяє розрізняти динамічні жести, в яких статична конфігурація кисті може збігатися, а відмінність полягає виключно в характері руху. Вікно спостереження становить 25 послідовних кадрів, що відповідає вхідному тензору форми (25, 126).

Порівнювались три архітектури нейронних мереж: однонаправлена LSTM, двонаправлена BiLSTM та одновимірна згорткова мережа 1D CNN. Рекурентні архітектури (LSTM та BiLSTM) мають однакову структуру: два рекурентні шари (128 та 64 нейрони відповідно), два повнозв’язних шари (256 та 128 нейрони з активацією ReLU) та вихідний шар із 33 нейронами та функцією активації softmax. Для регуляризації застосовано шари Dropout (коефіцієнти 0.25, 0.25, 0.3 та 0.2), BatchNormalization після кожного рекурентного та повнозв’язного шару, а також шар GaussianNoise (σ = 0.01) на вході для підвищення стійкості до тремтіння руки. Навчання виконувалось із використанням оптимізатора Adam, функції втрат sparse categorical

crossentropy, розміром пакету (batch size) 64 та ранньою зупинкою (patience = 16) за метрикою val_loss. Архітектура CNN мала такий вигляд: на вході мережі так само застосовано шар додавання гауссового шуму, за яким слідував блок екстракції ознак із двох послідовних згорткових шарів Conv1D на 64 та 128 фільтрів відповідно (розмір ядра – 3). Для регуляризації після кожного згорткового шару інтегровано шари BatchNormalization та Dropout (0.25). Після вирівнювання тензора шаром Flatten дані передавалися на повнозв'язний блок класифікації (Dense на 256 та 128 нейронів з активацією ReLU та Dropout 0.3 і 0.2 відповідно). Вихідний шар містив 33 нейрони з функцією активації softmax.

Для компенсації дисбалансу класів використано зважені коефіцієнти, обчислені методом balanced. Додатково скориговано вагу класів із систематично низькими показниками recall: для літери «Г» коефіцієнт збільшено у 1.9 рази, для «Ґ» – коефіцієнт знижено до 0.9, оскільки модель у процесі навчання була схильна класифікувати невизначені жести як «Ґ», замість «Г». Ці класи є найбільш проблематичними, оскільки їхня конфігурація кисті майже ідентична: для «Г» великий палець залишається нерухомим, тоді як для «Ґ» він виконує характерний рух, що ускладнює розрізнення навіть за наявності часового контексту.

Архітектура LSTM досягла точності 96.13% на тестовій вибірці, тоді як BiLSTM – 97.59%, 1D CNN досягла точності 98.22%. Показник macro F1-score для LSTM становив 0.96, для BiLSTM – 0.97, для 1D CNN – 0.98. Найбільші труднощі усі три моделі виявили на парах літер із подібною конфігурацією кисті: «Г» та «Ґ», що підтверджує необхідність включення швидкісних ознак та зважування класів для задач із малою міжкласовою відстанню в просторі ознак.

Висновки. У роботі проведено порівняльний аналіз архітектур LSTM, BiLSTM та 1D CNN для задачі розпізнавання жестів 33 літер української дактильної абетки на основі власного датасету відеопослідовностей з ключовими точками рук. Встановлено, що архітектура 1D CNN забезпечує вищу точність розпізнавання, що може бути пов'язане із ефективним вилученням локальних часових патернів через згорткові фільтри. Наукова новизна роботи полягає у використанні власного відеодатасету для української дактильної абетки з залученням кількох дикторів та умов освітлення, а також у систематичному порівнянні архітектур для цієї задачі. Практична цінність роботи полягає у відтворюваній методиці збору даних та порівняння архітектур, яка може бути застосована для інших малодосліджених жестових мов.

Список літератури

1. Lianyu Hu, Liqing Gao, Zekang Liu, Wei Feng. Dynamic Spatial-Temporal Aggregation for Skeleton-Aware Sign Language Recognition. arXiv. 2024. URL: https://arxiv.org/abs/2403.12519 (дата звернення: 15.05.2026).
2. Gerges H. Samaan, Abanoub R. Wadie, Abanoub K. Attia, Abanoub M. Asaad, Andrew E. Kamel, Salwa O. Slim, Mohamed S. Abdallah, Young-Im Cho. MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition. MDPI. 2022. URL: https://www.mdpi.com/2079-9292/11/19/3228 (дата звернення: 15.05.2026).
3. Deme Kuma Gonfa, Million Meshesha, Dagne Walle Girmaw, Kidane W/Maryam. A deep learning framework for Ethiopian sign language recognition using skeleton-based representation. Scientific Reports. 2025. URL: https://www.nature.com/articles/s41598-025-19937-0 (дата звернення: 15.05.2026).
4. Md. Zia Uddin, Costas Boletsis, Pål Rudshavn. Real-Time Norwegian Sign Language Recognition Using MediaPipe and LSTM. MDPI. 2025. URL: https://www.mdpi.com/2414-4088/9/3/23 (дата звернення: 15.05.2026).

Sergii Kuzmin
PhD Student at Department of Automated Control Systems, Lviv Polytechnic National University
Academic Supervisor Dr. Sci. (Eng), Professor OLEH BEREZSKY, Head of Computer Engineering Department, West Ukrainian National University

DOWNSTREAM CLASSIFICATION PROBE OF UNI2-H–CONDITIONED SYNTHETIC HISTOPATHOLOGY IMAGES

Introduction. The increasing use of synthetic histopathology images as substitutes or complements for the actual tissue patches in downstream training is driven primarily by two concerns. One is the lack of availability of sufficient numbers of well annotated and privacy cleared cohorts. The other is the need to be able to implement data augmentation pipelines with the type of single GPU computing environment typically found in university and clinical research settings. Published evaluations of these generators have generally been based on either a combination of distribution-based metrics (e.g., Fréchet Inception Distance (FID), Kernel Inception Distance (KID)), or the ability of the generated patches to augment existing realtraining sets resulting in improved classification accuracy. While both types of results are valuable, each also has some known limitations. FID and KID measures were developed using the Inception V3 feature space that was trained on natural images, and therefore these measures are poorly aligned to the morphologic characteristics important for classification of tissues. Secondarily, while data augmentation provides a single measure of improvement, this measure reflects a combination of three confounded factors (generator performance, patch generation strategy and downstream model regularization) and does not isolate the contribution of the generator by itself. Neither approach directly answers an even more basic question: how much class-discriminative information does the generator itself capture?

One possible way to directly measure how well we have captured this concept is by comparing the performance of a downstream classification model that has been trained only on our generated synthetic data versus the same model that has been trained only on the real training image data. The differences in performance can tell us approximately what the generator failed to capture relative to the real training image data. This is independent of any choices made regarding augmentation ratios, how to mix synthetic images with real ones, and/or regularization methods applied in a downstream study. That said, these types of probes are frequently omitted from papers describing histopathology synthesis systems. However, they are precisely what is needed to demonstrate that a parameter-efficient generator captures relevant domain information rather than simply generating samples that appear to be realistic but lack any informative content. Such probes also allow for the use of paired statistical testing techniques at multiple seeds and architectures, which yields a reproducible quantitative measure (i.e., percentage of class-discriminative signal preserved) that future downstream researchers can use to make comparisons.

In our prior work [1] we adapted Stable Diffusion 1.5 to colon histopathology by replacing CLIPbased text conditioning with embeddings from the UNI2-h pathology foundation model [2], inserted lowrank adapters (LoRA) [3] of rank 4 into the UNet attention blocks, and trained only a small projector together with the LoRA parameters (about 5.53 M trainable weights, or 0.64 % of the Stable Diffusion backbone). The pipeline reached a validation FID of 77.59 on the Chaoyang dataset [4] while running on a single 16 GB consumer GPU. The natural follow-up question, which the present work addresses, is whether the resulting synthetic images carry enough class-discriminative signal to train a competitive tissue classifier on their own.

Problem Statement. While prior studies showed improved results with respect to FID scores as well as augmentation gains, compared to baselines for histopathology generators, none of those studies directly assessed, with paired statistical testing across multiple seeds and architectures, how much of the classdiscriminative signal of a large collection of real-world histopathology images is preserved by a parameterefficient embedding-conditioned generator. The object of this research is the quality of colon tissue patch synthesis generated via UNI2-h–conditioned LoRA adaptation of Stable Diffusion 1.5. The subject is the direct downstream classification probe of training a tissue classifier with only synthetic or only real images. The goal of this study is to measure the gap between training a tissue

classifier on synthetic and on real images, and to identify per-class deviations that constrain the use of such generators in resource-limited augmentation pipelines.

Main Material. No modifications are made to the generative pipeline used in this work. It follows the same architecture described in [1], and uses the ablation checkpoint from [1] that produced the lowest validation FID. All layers of Stable Diffusion 1.5 remain frozen, while low-rank adapters of rank 4 are added to the query, key, value and output projections of every UNet attention block (both self-attention and crossattention). The CLIP text encoder is removed. Instead, we condition the model directly on UNI2-h, a selfsupervised pathology foundation model. UNI2-h produces a 1536-dimensional descriptor for each input patch. We project this descriptor into the 768-dimensional cross-attention token space of Stable Diffusion using a two-layer MLP with SiLU activation. The whole point of this setup is to skip text prompts altogether. The standard alternative, when adapting general text-to-image diffusion models to histopathology, is to write prompts that try to describe diagnostic patch content, which we avoid here. Training uses the standard noiseprediction objective on 512×512 Chaoyang patches, encoded into the 64×64 latent space of the frozen VAE. Only the projector and LoRA weights are updated. Generation uses the DDPM scheduler with 30 sampling steps and a classifier-free guidance scale of 1.3, which was found to be optimal for this embeddingconditioned configuration in [1].

In order to create a synthetic dataset with the exact same number of samples as our real Chaoyang dataset and the same per-class distribution, for every single training image (real) we generate a single synthesized image using our diffusion model conditioned on each patch's UNI2-h embedding. The only difference between the two datasets is whether the images are real or synthetic. We want the synthetic images to be reproducible, so we set the seed for each image in a deterministic way, following the convention of [1]. To normalize the stains for both real and synthetic images, we apply Macenko's stain normalization method [5] uniformly to all images. As opposed to selecting a reference image manually, we select it programmatically by identifying the training tile whose stain matrix is most similar to the median stain matrix calculated across all training tiles. This eliminates the selection bias from choosing a reference image as well as the staining variability.

It is also important to note that we performed slide-level splitting instead of the typical patch-level splitting utilized in many other histopathology benchmarks, with no slide or patient appearing in more than one partition. Slide-level splitting avoids leaking slide context into different partitions and tends to reduce overestimation of results. The LoRA generator itself was trained only on the Chaoyang training partition, and the held-out validation and test slides were never seen by the generator. So when we run the synth-only condition, we are really asking whether the class-discriminative information that survives the round-trip through the diffusion model is enough to classify real test-slide images, not whether the generator simply memorized its training set.

We utilized two backbone architectures, both initialized using ImageNet pretrained models. The first is ResNet-50 with the timm a1_in1k recipe [6]. The second is EfficientNet-B0 with the default timm pretraining. We used the same hyperparameters for both backbones. The optimizer was AdamW with a base learning rate of $3 \cdot 10^{-4}$ and weight decay of 10^{-4}. We used a linear warm-up over 2 epochs and then cosine decay down to 10^{-6}. The batch size was 32 and we trained in bfloat16 mixed precision. Input images were 256×256. The loss function was just unweighted cross-entropy. To isolate how differences in test performance were due to differences in training image sources rather than to extra training techniques, we did not utilize class weighting or any form of traditional image augmentation. Any differences in test performance can therefore be attributed to the source of the training images. For early stopping, we stopped training when the validation loss had not improved for ten epochs in a row. After that, we picked the checkpoint that had the lowest validation loss seen during training, and used that one for the test evaluation. Each combination of architecture and source (ResNet-50 / real, ResNet-50 / synth, EfficientNet-B0 / real, EfficientNet-B0 / synth) was evaluated over ten seeds {42, 123, 456, 789, 1024, 2048, 3141, 4096, 5555, 7777}, resulting in a total of forty unique training runs. Since the same seed values were used in the identical position across the different source conditions, the results could be paired for statistical comparison. Results are reported as averages ± standard deviations across seeds for each metric, and are compared using paired t-tests conducted on seed-matched pairs.

Experiments use the publicly available Chaoyang colon dataset [4], which contains four tissue classes (normal, serrated, adenocarcinoma, and adenoma) at 512×512 patch size. For the slide-level partition we used the same setup as our prior work [1]. We had 5 398 patches for training, 387 for validation, and 375 for the held-out test set. We kept the original class imbalance, so the training split had 1 914 adenocarcinoma patches, 1 573 normal, 1 032 serrated, and 879 adenoma. The class imbalance is intentionally not corrected: the goal of this work is to characterize the synth-only training condition under the same data conditions a downstream practitioner would actually face, not under an artificially balanced setup. Representative patches are shown in Figure 1, paired with synthetic samples drawn from the diffusion model conditioned on their UNI2-h embeddings.

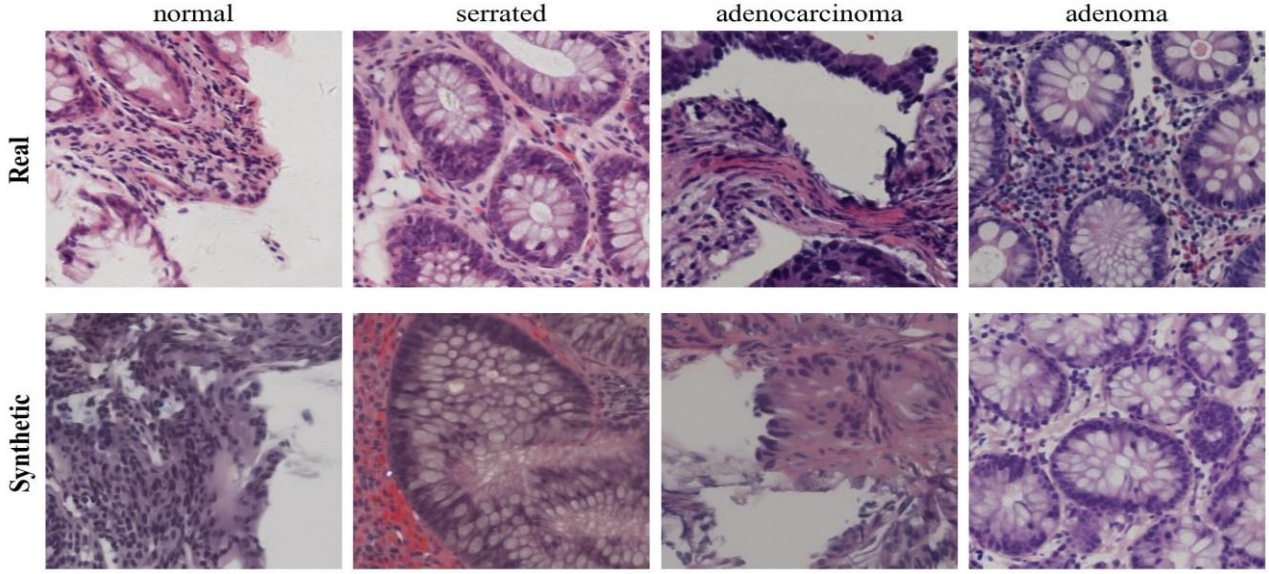


Figure 1. Real Chaoyang patches (top row) and corresponding synthetic patches drawn from the diffusion model conditioned on their UNI2-h embeddings (bottom row), one per class.

All experiments are executed on the same hardware used in [1], i.e., a Windows 11 workstation with an Intel Core i9-13980HX CPU, 32 GB of DDR5 RAM, and a single NVIDIA GeForce RTX 4090 Laptop GPU with 16 GB of VRAM, in order to remain consistent with a resource-constrained setting. Owing to early stopping, real-only runs converge in roughly 15 minutes and synth-only runs in roughly 17 minutes. The full forty-run probe completes in under twelve hours on this hardware, which is the same configuration that was used to train the generator in [1].

Table 1 shows the average test-set macro F1 and accuracy using the two architectures as well as the two training settings across all 10 seed experiments. With respect to the ResNet-50 architecture, we see that the synth-only training results in an average test macro F1 of 0.572 ± 0.054, while real-only training yields an average test macro F1 of 0.664 ± 0.034, which is a 13.8% difference between them. For the EfficientNetB0 architecture, we see that the corresponding numbers are 0.525 ± 0.044 for the synth-only training and 0.641 ± 0.018 for the real-only training, yielding an 18.2% difference. Each of these differences was found to be statistically highly significant when performing paired t-test comparisons, with each having very large effect size statistics. Accuracy at test also followed the same ordering. We can infer from this that the retained signal is contained within the 82%–86% interval across architectures. That is, a classifier trained solely on synthetic patches recovers more than 80% of the macro-F1 value associated with a classifier trained on the equivalent amount of real-world data, again far exceeding the random-chance value of 25% for a four-class classification task, and by a margin much larger than a generator producing visually believable yet class-uninformative samples would.

Table 1 - Test macro F1 and accuracy (mean ± std over 10 seeds) for real-only and synth-only training. Both gaps are significant at $p < 0.001$ under paired t-tests.

Architecture	Condition	Test macro F1	Test accuracy
ResNet-50	real_only	0.664 ± 0.034	0.782 ± 0.024
ResNet-50	synth_only	0.572 ± 0.054	0.716 ± 0.027
EfficientNet-B0	real_only	0.641 ± 0.018	0.751 ± 0.012
EfficientNet-B0	synth_only	0.525 ± 0.044	0.627 ± 0.051

The distribution of error for all four morphology types, as seen in Table 2, illustrates the fact that the error difference is not distributed evenly among all four morphology types. The majority type, adenocarcinoma, has an error difference of about 0.05 when moving from real to synthetic training on ResNet-50 and about 0.11 on EfficientNet-B0. The two minority types, serrated and adenoma, are impacted much more substantially. However, it is not the same minority class that suffers the largest drop on each architecture. On ResNet-50, adenoma is clearly the most affected. Its F1 score drops from 0.395 to 0.232, which is a relative decrease of 41%. On EfficientNet-B0, the picture changes. The F1 score for serrated drops from 0.495 to 0.356 (relative decrease of 28%), while the F1 score for adenoma drops from 0.400 to 0.308 (relative decrease of 23%). So, on this backbone, serrated rather than adenoma is the worst-affected class. However, both architectures indicate that synthetic-only training disproportionately affects the two rarest morphology types. We leave a more detailed mechanistic analysis to future work.

Table 2 - Per-class test F1 (mean over 10 seeds) for real-only and synth-only training, by architecture

Architecture	Condition	adenocarcinoma	normal	serrated	adenoma
ResNet-50	real_only	0.918	0.794	0.549	0.395
ResNet-50	synth_only	0.872	0.719	0.467	0.232
EfficientNet-B0	real_only	0.904	0.766	0.495	0.400
EfficientNet-B0	synth_only	0.797	0.636	0.356	0.308

Both architectures agree on the broad pattern: adenocarcinoma transfers well, while the two minority classes do not. This suggests that what we are seeing is a property of the synthetic data rather than something specific to one of the classifiers. The fact that an ImageNet-recipe ResNet-50 and an EfficientNet-B0 trained under the same hyperparameters end up agreeing is also a useful sanity check on the probe protocol itself.

There are some limitations to our current experimental setup that we would like to call attention to. Firstly, we only assess the probe with a single test dataset and with a single LoRA checkpoint. Assessing the probe with additional test data and using additional checkpoints for LoRA will provide us greater confidence in the generalizability of the retained-signal estimates. Secondly, since the probe only provides argmax-based macro F1 and accuracy as output values, we cannot determine if the missing signal occurs due to the confidence calibration or due to where the decision boundary is placed. We leave both of these extensions to a forthcoming broader study.

Conclusions. When we run a direct downstream probe of UNI2-h-conditioned LoRA-adapted Stable Diffusion, we find that the synthetic colon-tissue patches keep somewhere between 82% and 86% of the test macro-F1 performance of the real Chaoyang training data. This holds for both backbone architectures, across all ten seeds per condition, and under paired-t-test analysis. The retained fraction is large enough that we can argue the generator is capturing most of the class-discriminative structure of the source distribution, even though we only train about 5.53 M parameters out of 861 M, and even though everything runs on a single 16 GB consumer GPU at both the generation and the downstream-classification stages. The remaining gap is concentrated on the two minority classes, where relative F1 losses go up to 41% (adenoma on ResNet50) and up to 28% (serrated on EfficientNet-B0). This is a sign that synth-only training disproportionately affects rare morphologies on a class-imbalanced

dataset. Overall, these results tell us something about the quality of the embedding-conditioned generator as a source of class-discriminative training data, which is the necessary precondition for using it in any downstream augmentation. They also motivate a broader study that we are working on. In that study, we plan to extend the same protocol in two directions. First, to mixed real-and-synthetic training regimes, and second, to per-class generation strategies. What we are aiming for is to take the signal retention we have shown here and turn it into a measurable benefit for downstream classification.

References

1. S. Kuzmin, O. Berezsky. Development of a Parameter-Efficient Method for Biomedical Image Synthesis by Substituting Text Conditioning with Pathology Foundation Model Embeddings in Latent Diffusion. Technology Audit and Production Reserves: vol. 2, no. 2(88), 2026, pp. 66–75.
2. R. J. Chen, T. Ding, M. Y. Lu, D. F. K. Williamson, G. Jaume, A. Zhang, F. Mahmood. Towards a General-Purpose Foundation Model for Computational Pathology. Nature Medicine: vol. 30, 2024, pp. 850–862.
3. E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen. LoRA: Low-Rank Adaptation of Large Language Models. International Conference on Learning Representations (ICLR): 2022.
4. C. Zhu, W. Chen, T. Peng, Y. Wang, M. Jin. Hard Sample Aware Noise Robust Learning for Histopathology Image Classification. IEEE Transactions on Medical Imaging: vol. 41, no. 4, 2022, pp. 881–894.
5. M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, N. E. Thomas. A Method for Normalizing Histology Slides for Quantitative Analysis. IEEE International Symposium on Biomedical Imaging (ISBI): Boston, MA, USA, 2009, pp. 1107–1110.
6. R. Wightman, H. Touvron, H. Jégou. ResNet Strikes Back: An Improved Training Procedure in timm. NeurIPS Workshop on ImageNet: 2021.

Болкун Д. А.
студент 4 курсу кафедри АСУ, ІКНІ, НУ «Львівська політехніка»
Науковий керівник: асистент Глод С. І., кафедра АСУ, НУ «Львівська політехніка»

ВЕБОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ПІДТРИМКИ ПЕРСОНАЛЬНИХ ТРЕНУВАНЬ ІЗ АВТОМАТИЗОВАНИМ АНАЛІЗОМ ТЕХНІКИ ВИКОНАННЯ ВПРАВ

Вступ. Неправильна техніка виконання фізичних вправ є поширеною причиною травм як серед аматорів, так і серед досвідчених спортсменів [1]. За даними аналітичних досліджень, глобальний ринок фітнес-застосунків зростає з темпом понад 13% щорічно [2], однак переважна більшість наявних рішень обмежується кількісним обліком навантажень – підрахунком підходів, повторень і калорій – без оцінки якості виконання руху. Персональний тренер залишається єдиним доступним інструментом корекції техніки, проте його послуги фінансово недоступні для більшості користувачів. Це зумовлює потребу в автоматизованій системі аналізу техніки вправ у реальному часі.

Актуальність. Існуючі рішення для аналізу техніки вправ реалізовані переважно як нативні мобільні застосунки або вимагають встановлення окремого програмного забезпечення, що обмежує їхню кросплатформеність та доступність. Веборієнтована клієнт-серверна архітектура усуває ці обмеження: тонкий клієнт (React) відповідає виключно за відображення результатів та керування тренувальною сесією, тоді як захоплення кадрів відеопотоку та обчислювальна логіка аналізу пози виконується на серверній стороні засобами Python та MediaPipe Pose [3]. Такий розподіл забезпечує апаратну незалежність клієнтського пристрою, спрощує розгортання та оновлення серверної логіки без втручання користувача.

Постановка задачі. Метою роботи є розробка веборієнтованої інформаційної системи підтримки персональних тренувань із автоматизованим аналізом техніки виконання вправ. Система повинна забезпечувати:

- планування та ведення персонального журналу тренувань;
- аналіз техніки виконання вправ через відеопотік веб-камери у режимі реального часу;
- відстеження та відображення динаміки показників користувача за результатами тренувальних сесій.

Технічна проблема полягає в забезпеченні точності вимірювання кутів у цільових суглобах із похибкою не більше ±5° при виконанні вправ у фронтальній та сагітальній площинах при використанні стандартної веб-камери, рівномірному освітленні та відстані від камери 1,5–2,5 м без спеціалізованого обладнання.

Основна частина. Систему реалізовано за клієнт-серверною архітектурою, загальну структуру якої наведено на рис. 1. Клієнтська частина (React) функціонує як інтерактивний інтерфейс користувача, що забезпечує візуалізацію даних, отриманих від сервера, та керування параметрами тренувальної сесії. Серверна частина (FastAPI, Python) реалізує REST API для управління даними користувачів та бізнес-логікою застосунку. Ключовим технічним модулем сервера є підсистема обробки відеопотоку на базі бібліотеки OpenCV, яка здійснює захоплення кадрів, та модель MediaPipe BlazePose, що виконує виявлення ключових точок скелета в реальному часі. Журнал тренувань і профілі користувачів зберігаються у реляційній базі даних (MySQL), а взаємодія між рівнями здійснюється через HTTP-запити з JWT-автентифікацією.

Ключовим обчислювальним елементом системи є визначення кута α у цільовому суглобі на основі координат трьох суміжних ключових точок скелета: A (проксимальна точка), B (суглоб) та C (дистальна точка).

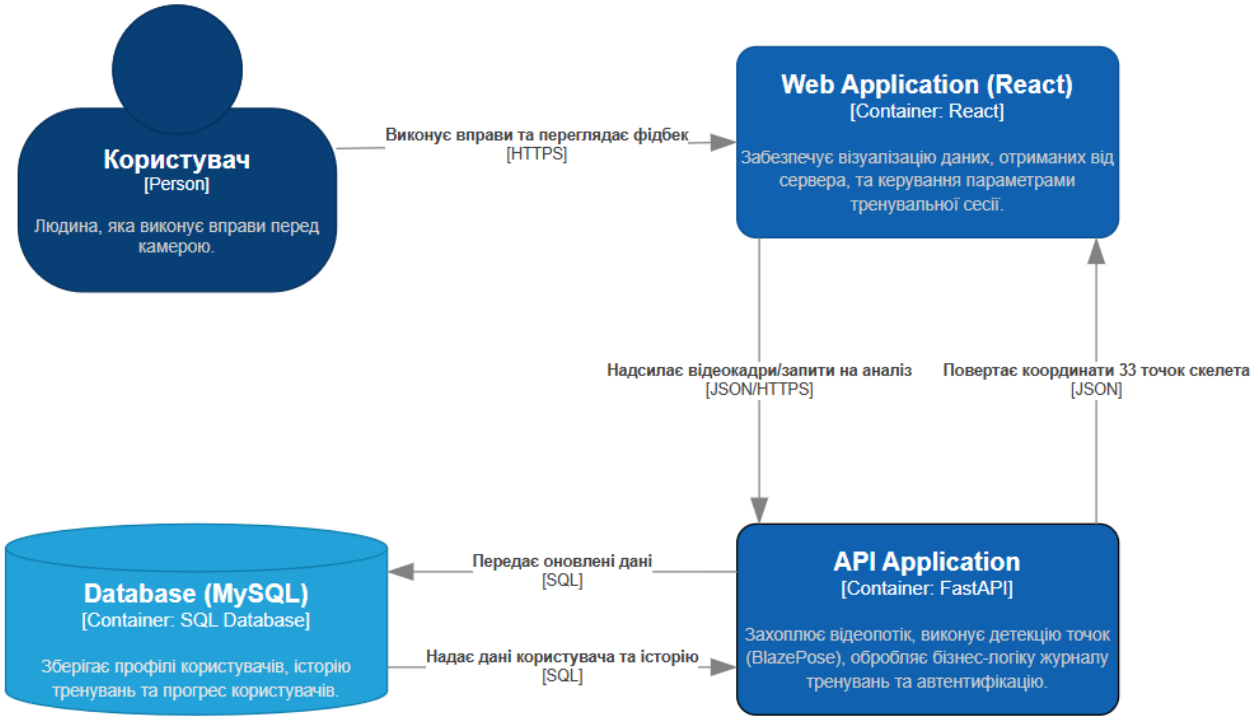


Рисунок 1 – *C4-діаграма архітектури системи*

Кут обчислюється через різницю кутів нахилу векторів $\overrightarrow{BA}$ та $\overrightarrow{BC}$ відносно горизонтальної осі за функцією arctan2:

$$\alpha = |arctan2(Cy - By, Cx - Bx) - arctan2(Ay - By, Ax - Bx)| \cdot (180 / \pi), \quad (1)$$

де arctan2(y, x) - арктангенсна функція, що повертає кут у радіанах в діапазоні (−π; π]; різниця кутів відображає кутовий зсув між напрямками $\overrightarrow{BA}$ та $\overrightarrow{BC}$.

Оскільки результат різниці може виходити за межі [0°; 180°], застосовується корекція:

$$\text{якщо } \boldsymbol{\alpha > 180°}: \boldsymbol{\alpha = 360° - \alpha}, \quad (2)$$

Це гарантує результат у діапазоні [0°; 180°], що відповідає анатомічному діапазону кутів у суглобах людини.

Поточне значення кута порівнюється з еталонним діапазоном для відповідної фази вправи: для присідання – кут у колінному суглобі 80–100° у нижній точці; для підйому на біцепс – кут у ліктьовому суглобі менше 50° у верхній точці; для віджимань від підлоги – кут у ліктьовому суглобі менше 90° у нижній точці. Зазначені еталонні значення ґрунтуються на фундаментальних засадах біомеханіки та спортивної медицини. Використання таких параметрів дозволяє забезпечити максимальну ефективність активації цільових м'язових груп при одночасній мінімізації ризику травми опорно-рухового апарату. Дані нормативи є результатом багаторічних кінезіологічних досліджень та підтверджені практичним досвідом у галузі професійного спорту [4]. На основі порівняння поточного кута з еталонним діапазоном система автоматично підраховує повторення та визначає фазу руху (опускання/підйом). Порівняльний аналіз розробленої системи з наявними аналогами за ключовими функціональними критеріями наведений у табл. 1.

Як видно з табл. 1, розроблена система є єдиним безкоштовним рішенням, що одночасно підтримує аналіз техніки, доступ через веббраузер без встановлення ПЗ та ведення журналу тренувань. Kemtai є найближчим аналогом за функціональністю, однак потребує платної підписки та не зберігає журнал тренувань. Kaia Health і Onyx Coach реалізовані виключно як нативні мобільні застосунки, що обмежує їхню кросплатформеність. Таким чином, більшість існуючих рішень або обмежені у функціональності, або недоступні для широкого кола

користувачів через платну модель чи вимогу нативної установки, що підтверджує доцільність розробки запропонованої системи.

Таблиця 1. Порівняльний аналіз розробленої системи з наявними рішеннями

Рішення	Аналіз техніки	Веб-браузер	Без спец. обладнання	Журнал тренувань	Доступ
Kemtai	✓	✓	✓	✗	Платний
Kaia Health	✓	✗	✓	✓	Платний
Onyx Coach	✗	✗	✓	✓	Платний
Hevy	✗	✓	✓	✓	Безкоштовний
Розроблена система	✓	✓	✓	✓	**Безкоштовний**

Тестування прототипу проводилося для 14 базових вправ (рис. 2). Умови тестування: стандартна веб-камера 1080p, рівномірне штучне освітлення, відстань 1,5–2,5 м від камери, фронтальна або сагітальна площина знімання. За таких умов модель виявлення пози BlazePose (що входить до складу MediaPipe Pose) забезпечує виявлення ключових точок із частотою понад 25 кадрів на секунду, а середня похибка вимірювання кута у цільовому суглобі не перевищує ±5°. Система має задокументовані обмеження: при недостатньому освітленні, значних відхиленнях від фронтальної або сагітальної площини, а також при частковому перекритті кінцівок – точність виявлення знижується, оскільки використовується лише стандартна веб-камера без глибинного сенсора.

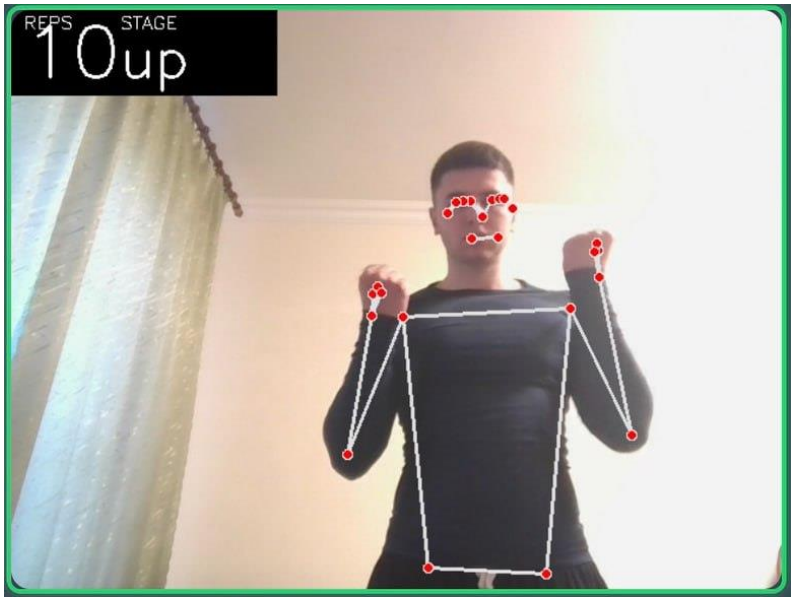


Рисунок 2 – Прототип системи під час виконання вправи: відображення скелетної моделі, лічильника повторень та поточної фази руху

Логіку взаємодії користувача з системою унаочнює діаграма прецедентів (рис. 3). Користувач отримує доступ через реєстрацію та авторизацію, після чого може переглядати каталог вправ, формувати персональні плани та відстежувати прогрес через журнал тренувань. Центральним сценарієм є запуск аналізу техніки: система виконує виявлення ключових точок скелета, обчислює кути у цільових суглобах, верифікує техніку за біомеханічними еталонами та відображає скелетну модель поверх відеопотоку. Результати кожної сесії автоматично зберігаються у базі даних.

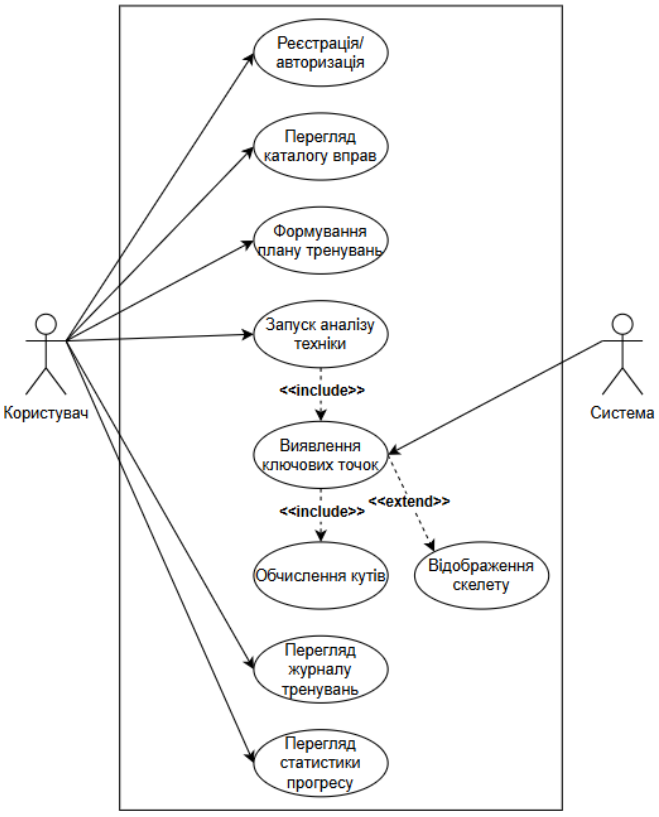


Рисунок 3 – Use case *діаграма системи*

Висновок. У роботі представлено веборієнтовану інформаційну систему підтримки персональних тренувань із автоматизованим аналізом техніки виконання фізичних вправ у режимі реального часу. Реалізовано клієнт-серверну архітектуру, в якій серверний модуль на основі MediaPipe BlazePose виконує розпізнавання 33 ключових точок скелета та обчислення кутів у цільових суглобах із похибкою до ±5° при частоті обробки понад 25 кадрів на секунду без залучення спеціалізованого обладнання. Система підтримує автоматичний підрахунок повторень, визначення фази руху та ведення персонального журналу тренувань. Порівняльний аналіз засвідчив, що розроблене рішення є єдиним безкоштовним веборієнтованим застосунком серед розглянутих аналогів, що поєднує аналіз техніки виконання вправ із журналом тренувань. Перспективою подальшого розвитку є розширення каталогу підтримуваних вправ, підтримка сагітальної площини знімання та інтеграція глибинного сенсора для підвищення точності розпізнавання в складних умовах освітлення.

Список літератури

1. Подоляка П. С., Ногас А. О., Гуцман С. В., Андреєва О. Б. Спортивний травматизм у сучасному спорті. 2022. URL: https://rehabrec.org/index.php/rehabilitation/article/view/237.

2. Grand View Research. Fitness App Market Size, Share & Trends Analysis Report. 2023. URL: https://www.grandviewresearch.com/industry-analysis/fitness-app-market.

3. Bazarevsky V., Grishchenko I., Raveendran K., Zhu T., Zhang F., Grundmann M. BlazePose: On-device Real-time Body Pose tracking. arXiv:2006.10204. 2020.

4. Коваль В. В., Чернозуб А. А., Альошина А. І. Проблема оптимізації техніки виконання вправ у силовому фітнесі, враховуючи фізіологічні процеси адаптації осіб різних вікових груп до навантажень. 2024. URL: https://dspace.uzhnu.edu.ua/items/ab86429d-ef72-46e9-af7e-d611547cfe41.

Гаврилюк В.А.
студент 3 курсу АСУ, ІКНІ, НУ ЛП
Науковий керівник д.т.н., професор Теслюк В.М., кафедра АСУ, НУ ЛП

ПІДХІД ДО АНАЛІЗУ ЗМІН ІНФРАСТРУКТУРИ ЯК КОДУ ЗАСОБАМИ ІНТЕРАКТИВНОЇ ГРАФОВОЇ ВІЗУАЛІЗАЦІЇ

Вступ. Сучасні підходи до розроблення програмного забезпечення передбачають часте внесення змін до хмарної інфраструктури. Дослідження DORA (DevOps Research and Assessment) демонструють пряму залежність між частотою розгортань і стабільністю систем: команди найвищої продуктивності здійснюють розгортання кілька разів на день, досягаючи при цьому вищої надійності порівняно з командами, що впроваджують зміни рідше [1]. Досягнення такої частоти потребує гнучкої, відтворюваної інфраструктури, що реалізується через підхід інфраструктури як коду (Infrastructure as Code, IaC) – формалізацію опису інфраструктури у вигляді декларативного коду, який застосовується для автоматизованого розгортання та модифікації систем.

Серед IaC-інструментів широкого поширення набув Terraform компанії HashiCorp – незалежне від платформ рішення, що дозволяє описувати ресурси будь-якого хмарного провайдера мовою HCL (HashiCorp Configuration Language). Перед застосуванням змін Terraform формує детальний план, що перераховує всі заплановані дії щодо ресурсів інфраструктури. Зі зростанням масштабів систем аналіз таких планів стає нетривіальним завданням, що безпосередньо впливає на якість рецензування змін і надійність інфраструктури

Постановка задачі. Об'єкт дослідження – процес аналізу та верифікації планів змін інфраструктури як коду. Предмет дослідження – метод інтерактивної графової візуалізації планів Terraform як засіб підвищення наочності та інтерпретованості змін інфраструктури. Головна мета даного дослідження полягає у розробленні підходу до візуалізації планів Terraform у вигляді інтерактивного графа залежностей на основі внутрішньої DAG-моделі (directed acyclic graph, спрямований ациклічний граф) інструменту, що забезпечує наочне відображення ресурсів, їхніх зв'язків та запланованих станів змін, і спрощує процес рецензування інфраструктурних змін у командах.

Основний матеріал. Традиційним підходом до верифікації планів Terraform залишається рецензування текстового звіту колегами [2]. Лінійний формат плану не відображає залежностей між ресурсами, а зі зростанням їх кількості звіт перетворюється на «стіну тексту», що спричиняє когнітивне навантаження та ускладнює оцінку фактичного впливу змін на систему [3]. Підвищене когнітивне навантаження є особливо критичним у контексті інфраструктурних змін, оскільки тестування інфраструктури залишається обмеженим, повільним і економічно витратним порівняно з тестуванням програмного коду [4], а наслідки непомічених помилок – недоступність сервісів, втрата даних або порушення безпеки – можуть мати значний негативний вплив на функціонування системи. Проблема загострюється в умовах багатофункціональних команд, де не всі інженери володіють однаковим рівнем знань інструментів інфраструктури, а інфраструктурний код у декларативній формі суттєво відрізняється від типового програмного коду. Для таких учасників інтерпретація плану є ще складнішою та вимагає додаткових пояснень, що уповільнює процес затвердження змін і частково нівелює основні переваги IaC [5].

Існуючі підходи до вирішення цієї проблеми можна поділити на чотири категорії: інструменти покращеного текстового представлення, генератори коду через діаграми, системи інтеграції у процеси рецензування та інструменти графічного представлення планів. Перша і третя категорії успадковують фундаментальні недоліки текстового виводу, друга - змінює парадигму керування інфраструктурою і вимагає переосмислення усталених підходів до розробки. Саме четверта категорія є предметом подальшого аналізу.

Графічне представлення залежностей є усталеним методом зниження когнітивного навантаження при аналізі складних систем [3], і у випадку Terraform воно є особливо природним: інструмент внутрішньо будує направлений ациклічний граф залежностей між

ресурсами як основу планування. Проте існуючі інструменти графічного представлення планів не використовують цей потенціал повною мірою.

У таблиці 1 наведено Порівняння існуючих інструментів візуалізації змін

Таблиця 1 - Порівняння існуючих інструментів візуалізації змін

Критерії порівняння	Terraform Graph	Terraform-visual	Rover	Pluralith
Тип візуалізації	Статичний граф	Деревоподібна структура	Інтерактивний граф	Інтерактивний граф
Відображення зв'язків	+	-	+	+
Відображення стану змін	-	+	+	+
Інтерактивність	-	(фільтрування)	(масштабування)	+
Автономність	Так	Так	Так	Ні(SaaS)
Актуальність підтримки	+	+	+	-
Формат виводу	Формат файлу GraphViz	HTML	Веб-інтерфейс	Веб-інтерфейс / настільний застосунок

Аналіз виявляє незайняту нішу: жоден з активно підтримуваних інструментів не поєднує одночасно повноцінне відображення залежностей між ресурсами, індикацію запланованих станів змін, розвинену інтерактивність та централізований командний перегляд. Єдиний функціонально повний інструмент – Pluralith – є недоступним через призупинення розробки, а його архітектура додатково створювала ризики інформаційної безпеки через обробку планів на інфраструктурі постачальника. Окремо варто зазначити, що більшість актуальних інструментів орієнтовані на локальне використання і не передбачають централізованого перегляду планів, що є суттєвим обмеженням для команд з високим рівнем співпраці, де перегляд планів колег вимагає розгортання власної локальної копії застосунку.

Для усунення виявлених прогалин необхідний підхід, що поєднує графову модель залежностей з продуманою нотацією та інтерактивністю. Втім, наявність графового представлення сама по собі не гарантує ефективності – безсистемне використання візуальних елементів може вводити в оману та призводити до неправильного тлумачення змін. Ефективна візуалізація планів має забезпечувати кольорове кодування запланованих станів ресурсів, відображення залежностей як спрямованих ребер графа та механізми фільтрування і деталізації для керування складністю великих планів. Оскільки рецензування інфраструктурних змін залучає учасників з різним рівнем технічної експертизи – від інженерів інфраструктури до менеджменту, що здійснює фінальне погодження – централізований перегляд з локальною обробкою даних є необхідною умовою як для ефективної командної співпраці, так і для безпеки чутливої інформації, що міститься у планах.

Висновки. Аналіз існуючих підходів до інтерпретації планів змін інфраструктури як коду підтвердив недостатність текстового представлення для повноцінного рецензування в умовах зростаючої складності систем та багатофункціональних команд. Застосування інтерактивної графової візуалізації на основі DAG-моделі Terraform дозволяє усунути невідповідність між внутрішньою моделлю інструменту та зовнішнім представленням плану, забезпечуючи наочність залежностей між ресурсами та запланованих станів змін. Запропонований підхід, на відміну від існуючих аналогів, поєднує розвинену інтерактивність, централізований командний перегляд та локальну обробку даних, що знижує когнітивне навантаження при рецензуванні та зменшує ймовірність помилкових дій при застосуванні планів.

Список джерел

1. Google Cloud. (2024). 2024 accelerate state of devops. DevOps Research and Assessment (DORA). https://dora.dev/research/2024/dora-report/2024-dora-accelerate-state-of-devops-report.pdf

2. Nicole, F., Jez, H., & Gene, K. (2018). Accelerate building and scaling high performing technology organizations by nicole forsgren jez humble gene kim. IT Revolution Press.

3. Hermans, F., & Skeet, J. (2021). The Programmer's brain: what every programmer needs to know about cognition. Manning.

4. Sokolowski, D. (2022). Infrastructure as code for dynamic deployments. У Proceedings of the 30th ACM joint european software engineering conference and symposium on the foundations of software engineering (с. 1775–1779). ACM. https://doi.org/10.1145/3540250.3558912

5. Leite, L., Kon, F., Pinto, G., & Meirelles, P. (2020). Building a theory of software teams organization in a continuous delivery context. У Proceedings of the ACM/IEEE 42nd international conference on software engineering: Companion proceedings (с. 296–297). ACM. https://doi.org/10.1145/3377812.3390807

Гілета А.Р., Кришевська В. П.
магістрантки 1 курсу кафедри АСУ, ІКНІ, Національний університет "Львівська політехніка"
Науковий керівник к.т.н., професор Різник В.В., кафедра АСУ, ІКНІ, Національний університет "Львівська політехніка"

ОСОБЛИВОСТІ ПОПЕРЕДНЬОЇ ОБРОБКИ ТА ВЕКТОРИЗАЦІЇ ТЕКСТОВИХ ДАНИХ СОЦІАЛЬНИХ МЕРЕЖ У СИСТЕМАХ АНАЛІЗУ ТОНАЛЬНОСТІ

Вступ. Експоненційне зростання обсягів неструктурованих даних у соціальних мережах, зокрема у Twitter (X), створює нові можливості для соціологічних та маркетингових досліджень. Проте автоматизований аналіз тональності або ж Sentiment Analysis таких текстів стикається з низкою проблем, зумовлених специфікою інтернет-комунікації. Тексти мікроблогів характеризуються обмеженою довжиною, порушенням граматичних норм, використанням специфічного сленгу та емодзі [1].

Аналіз останніх фахових публікацій свідчить про те, що навіть застосування найпотужніших архітектур глибокого навчання, таких як трансформери або гібридні нейронні мережі, не забезпечує стабільної точності класифікації за умови ігнорування етапу препроцесингу. Надмірна кількість лексичного «шуму» та технічних артефактів спотворює векторне представлення тексту, що призводить до помилкової інтерпретації сентименту [2]. Таким чином, актуальність роботи зумовлена необхідністю розробки та теоретичного обґрунтування оптимального конвеєра підготовки даних. Такий конвеєр має забезпечити ефективну фільтрацію технічного баласту, одночасно зберігаючи семантичне ядро та емоційне забарвлення повідомлення для подальшої обробки моделями машинного навчання.

Актуальність роботи полягає у визначенні оптимального конвеєра підготовки даних, який забезпечує збереження семантичного ядра повідомлення при максимальному видаленні технічного баласту.

Постановка задачі. У межах дослідження необхідно вирішити проблему ефективної трансформації зашумлених текстових даних у структурований числовий формат.

Об’єктом дослідження є технологічні процеси інтелектуального опрацювання та аналізу неструктурованої текстової інформації, отриманої з відкритих джерел соціальних медіа.

Предметом дослідження виступають методи та алгоритми багаторівневої попередньої обробки, програмні засоби фільтрації артефактів скрейпінгу, а також математичні моделі векторного представлення лексем для їх подальшої класифікації.

Метою роботи є розробка комплексного алгоритмічного забезпечення та детальне описання етапів функціонування конвеєра обробки даних (Data Preprocessing Pipeline). Основний акцент зроблено на поєднанні детермінованих методів (використання регулярних виразів для семантичного очищення), методів лінгвістичної нормалізації та багаторівневої векторизації (статистичної та семантичної), що дозволить уніфікувати підготовку даних як для класичних статистичних алгоритмів, так і для сучасних нейромережевих моделей.

Основний матеріал. Для дослідження використано еталонний датасет Sentiment140 [3], що містить 1,6 млн записів. Розвідувальний аналіз даних (EDA) виявив, що початкова структура містить надлишкові атрибути (ids, date, flag, user), які були вилучені для оптимізації пам'яті. Аналіз текстового поля показав наявність значного «шуму»: HTML-сутностей (наприклад, токен "quot"), URL-адрес та специфічного інтернет-сленгу.

У межах системи «Insight AI» реалізовано багаторівневий конвеєр попередньої обробки: технічне очищення (RegEx), обробка емодзі (перетворення графічних символів у текст) та лінгвістична нормалізація (Case Folding, Stop-words removal). Особливу увагу приділено скороченню подовжених слів ("goooood" → "good"), що дозволило зменшити розмірність словника на 15%.

Центральним етапом підготовки є векторизація тексту. Оскільки алгоритми машинного навчання працюють виключно з числовими тензорами, векторизація є необхідним містком між природною мовою та математичним апаратом моделі. Вона перетворює дискретні слова у точки багатовимірного простору, де математична відстань між векторами відображає ступінь їх семантичної або статистичної подібності [4].

У роботі використано два паралельні підходи векторизації, що зумовлено архітектурними особливостями обраних моделей:

1. Статистичний підхід (TF-IDF): призначений для класичних алгоритмів (логістична регресія, SVM). Він оцінює важливість слова на основі його частоти в конкретному тексті відносно всього корпусу. Додавання біграм дозволяє враховувати локальний контекст (наприклад, розрізняти "not" і "not happy").

2. Семантичний підхід (Word Embeddings / GloVe): розроблений для глибокого навчання (Bi-LSTM). На відміну від TF-IDF, де кожне слово є незалежним виміром, GloVe (див. рис. 1) розміщує слова у щільному 300-вимірному просторі, зберігаючи складні семантичні зв'язки (синонімію, асоціації) [5].

```
embedding_matrix = np.zeros((vocab_size, EMBEDDING_DIM))
for word, i in word_index.items():
  embedding_vector = embeddings_index.get(word)
  if embedding_vector is not None:
    embedding_matrix[i] = embedding_vector
```

```
embedding_layer = tf.keras.layers.Embedding(vocab_size,
                                            EMBEDDING_DIM,
                                            weights=[embedding_matrix],
                                            input_length=MAX_SEQUENCE_LENGTH,
                                            trainable=False)
```

Рис. 1. Програмна реалізація шару вбудовування (Embedding Layer) з використанням попередньо навчених векторів GloVe

Для порівняння характеристик цих методів у таблиці 1 наведено їхні ключові відмінності.

Таблиця 1. Порівняльний аналіз методів векторизації тексту

Характеристика	TF-IDF з N-gram	Word Embeddings (GloVe)
Тип представлення	Розріджені вектори великої розмірності	Щільні вектори фіксованої розмірності
Облік контексту	Тільки сусідні слова (через N-грами)	Глибокий семантичний контекст
Розмірність	Залежить від розміру словника (десятки тисяч)	Фіксована (300 компонентів)
Основна перевага	Швидкість обчислення, інтерпретованість	Здатність до узагальнення синонімів
Цільова модель	Classic ML (LR, SVM, Naive Bayes)	Deep Learning (CNN, LSTM, BERT)

На основі аналізу розподілу довжин повідомлень було встановлено порогове значення MAX_SEQUENCE_LENGTH = 30. Оскільки 98% твітів у корпусі не перевищують цю довжину (див. рис. 2), такий вибір дозволяє уникнути надмірного використання операції доповнення (padding) нулями, що підвищує обчислювальну ефективність нейронної мережі.

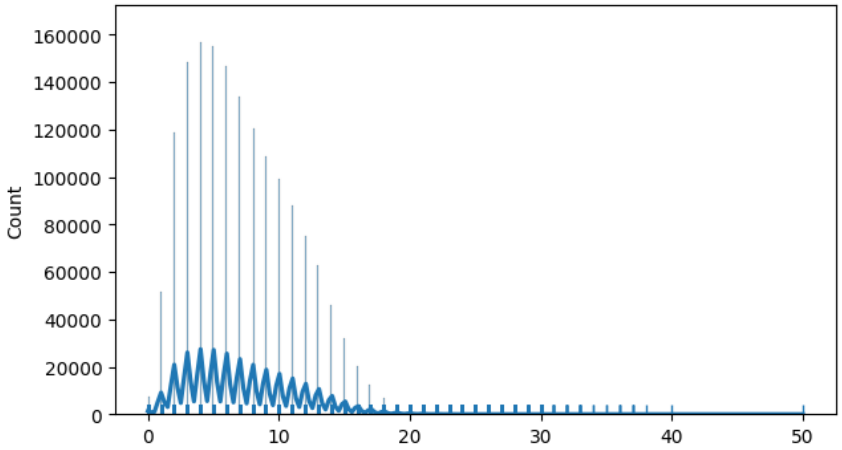


Рисунок 2. Гістограма розподілу довжини повідомлень

Архітектурно модуль обробки (Processing Module) інтегрований у багатошарову структуру системи як посередник між інтерфейсом на Streamlit та обчислювальним ядром, забезпечуючи перетворення «сирого» вводу користувача у стандартизовані вектори згідно з обраною моделлю аналізу. На рис. 3 зображено результат роботи описаного модуля.

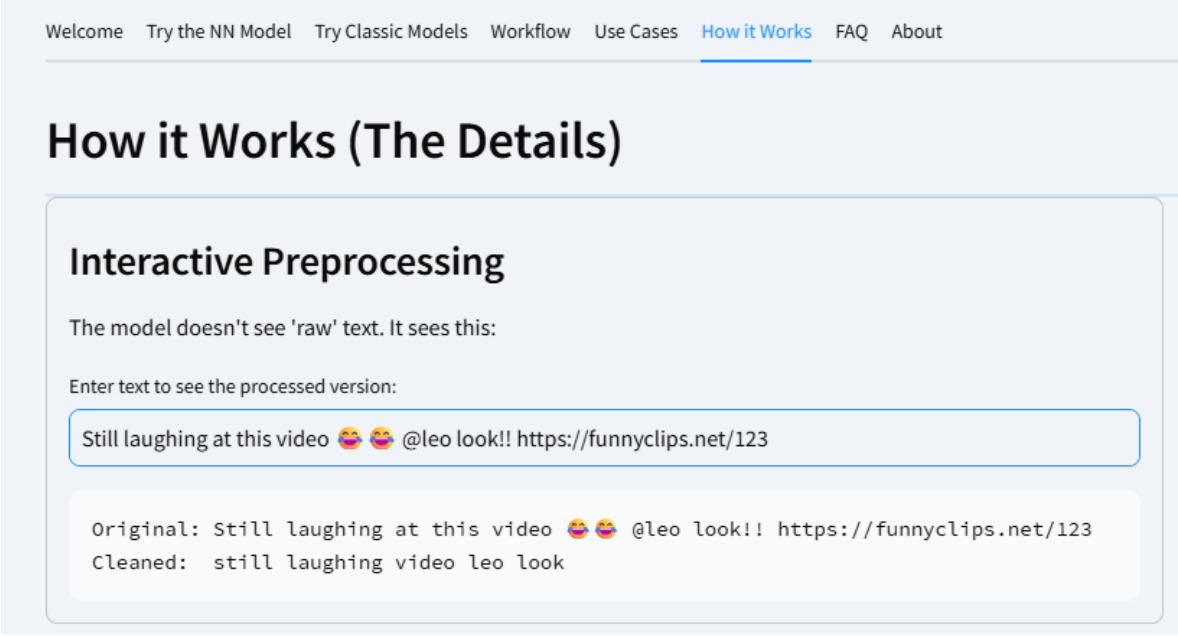


Рисунок 3. Ілюстрація роботи алгоритму попередньої обробки тексту

Висновки. У ході дослідження було розроблено та протестовано комплексний метод підготовки текстових даних соціальних мереж. Встановлено, що поєднання семантичного очищення (обробка емодзі, виправлення повторів) та комбінованої векторизації дозволяє суттєво підвищити якість вхідних даних.

Практична цінність результатів полягає у можливості використання запропонованого конвеєра для широкого спектру задач NLP, де робота ведеться з короткими, зашумленими текстами. Наукова новизна полягає в оптимізації параметрів векторизації спеціально для українського та англомовного сегментів Twitter з урахуванням специфіки мікроблогінгу.

Список літератури

1. Suryawanshi, N. S. (2024). Sentiment analysis with machine learning and deep learning: A survey of techniques and applications. *International Journal of Science and Research Archive*, 12(2), 005–015. https://doi.org/10.30574/ijsra.2024.12.2.1205
2. Raza, A., Jha, C. L., & Shahi, G. G. (2022). BERT-Based Approach for Sentiment Analysis of Social Media Data. *IEEE Access*, 10, 25969–25980. https://doi.org/10.1109/ACCESS.2022.3155110
3. Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(12). https://www-cs.stanford.edu/people/alecmgo/papers/twitter_nlp3.pdf
4. Nugroho, H., Tjahyanto, A., & Arifin, A. Z. (2023). A Hybrid CNN-BiLSTM Model with Metaheuristic Optimization for Sentiment Analysis. *IEEE Access*, 11, 90467–90479. https://doi.org/10.1109/ACCESS.2023.3302220
5. Al-Maitah, M., Ababneh, A., & Jararweh, Y. (2021). A Comparative Study of Machine Learning and Deep Learning Techniques for Sentiment Analysis. *Journal of Big Data*, 8(1), 1–20. https://doi.org/10.1186/s40537-021-00444-z

---

Глод Сергій Ігорович
аспірант 3 курсу кафедри АСУ, ІКНІ, НУ «Львівська Політехніка»
Науковий керівник к.т.н., доц. Дорошенко А. В.

EDGE-DEPLOYED YOLO26 FOR REAL-TIME MILITARY OBJECT DETECTION

Introduction. Modern conflicts have made real-time object detection a core requirement of military intelligence systems. Widespread deployment of UAVs, USVs, and UGVs produces continuous high-volume video and sensor streams that exceed the capacity of human analysts to monitor in real time, particularly for small, camouflaged, or partially occluded targets in cluttered scenes. At the same time, contested electromagnetic environments – including GPS spoofing and jamming of command links – make reliance on cloud-based inference impractical [1]. Detection must therefore run on the platform itself, on size-, weight-, and power-constrained embedded hardware, with low and predictable latency. These constraints define the design target: a detector that combines competitive accuracy with deterministic, low-latency inference on edge devices such as the NVIDIA Jetson Orin Nano. Ultralytics YOLOv26 (also known as YOLO26), released in January 2026, is built explicitly for this profile. It introduces a native end-to-end NMS-free detection head, removes Distribution Focal Loss to simplify export, and introduces the ProgLoss objective and Small-Target-Aware Label Assignment (STAL) rule, which target small-object accuracy – a known weak point of earlier YOLO generations and a recurring failure mode in aerial military imagery. This work evaluates YOLO26 on a custom military object detection dataset of three armored vehicle classes (tanks, BMPs, BTRs). It characterizes its accuracy and inference speed on the Jetson Orin Nano under conditions representative of real-world deployment.

Problem Statement. Real-time military object detection at the tactical edge faces a recurring failure mode: low recall for small objects [4]. Targets in aerial and long-range surveillance imagery frequently occupy under 1% of the frame area, and in the dataset used in this study, the 0-1% size bin is the dominant category for tanks and BMPs. Under standard Task Alignment Learning (TAL) and fixed-IoU anchor assignment, objects whose bounding boxes fall below roughly 8 pixels in a 640-pixel input often receive no positive anchor assignments at all, contributing no supervisory signal during training and producing systematically low recall at inference. YOLO26 addresses this directly. Its Small-Target-Aware Label Assignment (STAL) modifies the training-time matching rule so that objects below the 8-pixel threshold (relative to a 640×640 input) receive at least 4 assigned anchors regardless of IoU, ensuring that small targets contribute to the loss in every batch they appear in. This is paired with Progressive Loss Balancing (ProgLoss), which reweights loss components across training to prevent large or easy objects from dominating later epochs.

This work aims to evaluate whether YOLO26, through its STAL and ProgLoss training mechanisms, achieves improved small-object detection performance on a domain-specific military dataset while preserving real-time inference capability on resource-constrained edge hardware. To achieve this aim, the following objectives are defined:

1. To fine-tune YOLO26s on a custom military object detection dataset using a reproducible training protocol with documented hyperparameters.

2. To evaluate detection performance using standard metrics (precision, recall, mAP@0.5, mAP@[0.5:0.95]) with per-class breakdown.

3. To benchmark inference latency and throughput on the NVIDIA Jetson Orin Nano Super, confirming that accuracy gains are compatible with real-time edge deployment.

Main part. YOLO26 builds on the single-stage detection paradigm of its predecessors but introduces four targeted changes aimed at deployment on edge hardware. Architecturally, it retains the C3k2 CSP bottleneck blocks and the C2PSA spatial-attention module introduced in YOLOv11, with the SPPF block in the neck augmented by a shortcut connection that improves information flow between the input and the pooled output. The detection head retains three multi-scale outputs for small, medium, and large objects. The most consequential change is the elimination of Non-Maximum Suppression. During training, YOLO26 uses a dual-head configuration: a one-to-many head provides dense supervision to the backbone and neck, while a one-to-one head directly predicts a single bounding box per object. At inference, the one-to-many head is disabled, and detections are produced only by the one-to-one head, removing NMS post-processing entirely. This both simplifies export to ONNX, TensorRT, and TFLite and yields up to 43% faster CPU inference relative to YOLOv11n at comparable accuracy. Distribution Focal Loss, used in YOLOv8 through YOLOv11 to represent box coordinates as a discrete distribution, is removed in favor of direct regression. This reduces head complexity and improves compatibility with INT8/FP16 quantization for edge runtimes. Two training-time mechanisms address common YOLO failure modes. Progressive Loss Balancing (ProgLoss) reweights the relative contributions of classification, localization, and head-specific losses across training epochs, emphasizing one-to-many supervision early and gradually shifting toward one-to-one head as training stabilizes. Small-Target-Aware Label Assignment (STAL) ensures positive anchor assignments for objects below the 8-pixel threshold and is the primary mechanism by which YOLO26 is expected to improve the detection of small armored vehicles in this study. YOLO26 replaces the default SGD/AdamW choice with MuSGD, a hybrid of SGD and the Muon optimizer originally developed for large language model training. MuSGD applies Muon-style orthogonalized updates to selected parameter groups while keeping SGD-style updates elsewhere, aiming to achieve more stable late-epoch convergence [2-3]. Taken together, these four changes: NMS-free inference, DFL removal, ProgLoss with STAL, and the MuSGD optimizer – position YOLO26 as an edge-first detector with explicit small-object training mechanisms, which makes it a direct match for the dataset used in this study.

The dataset consists of approximately 12,000 images annotated for three armored vehicle classes: tanks, BMPs (infantry fighting vehicles), and BTRs (armored transporters). Frames were extracted from publicly available battlefield and reconnaissance videos, deduplicated, and resized to a uniform resolution while preserving aspect ratios. Annotation was performed in Roboflow and exported in YOLO TXT format. The dataset is split 77/13/10 across training, validation, and test partitions with class proportions preserved.

The distribution of object area relative to image area (Fig. 1) is heavily skewed toward small objects: the 0-1% bin is dominant for all three classes (~16% of tank and ~15% of BMP instances), and the combined 0-3% range accounts for more than half of all annotations. Small targets in this size range represent a well-documented failure mode for conventional detectors and are the specific case STAL is designed to address.

The spatial distribution of bounding-box centers and the joint distribution of box width and height (Fig. 2) show that objects span the full image frame with moderate central concentration. The small box dimensions dominate both axes – most annotations occupy less than 20% of the image width and height. The three classes overlap substantially in both center location and size, so the detector cannot rely on positional or scale priors and must learn discriminative appearance features.

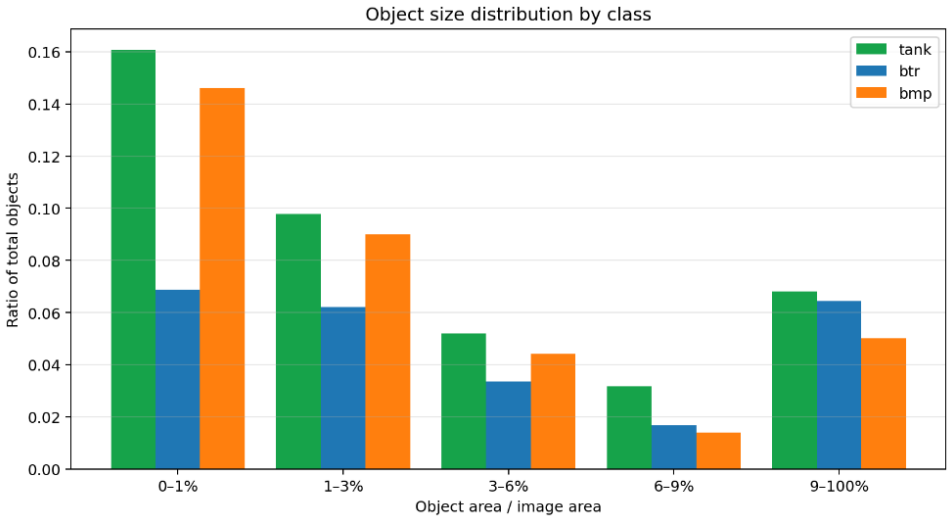


Fig. 1. Object size distribution by class, expressed as the ratio of bounding-box area to image area across five size bins

Having established the dataset's small-object profile, we now describe the training configuration used for YOLO26s and report its detection performance on the held-out test set.

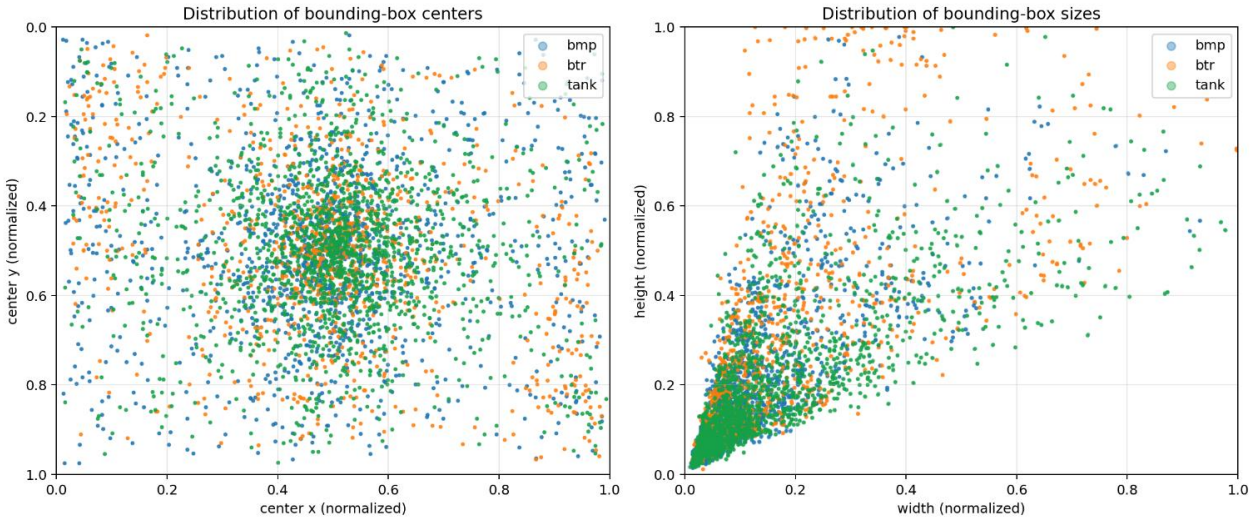


Fig. 2. Spatial distribution of bounding-box centers (left) and joint distribution of normalized box width and height (right), colored by class.

The YOLO26s variant (9.47M parameters, 20.5 GFLOPs) was selected over the smaller YOLO26n to retain sufficient capacity for the three-class fine-grained distinction (tank, BMP, BTR) while keeping the parameter count low enough for Jetson-class deployment. Training was performed on an NVIDIA RTX PRO 6000 Blackwell GPU using the Ultralytics framework (v8.4.51) with PyTorch 2.10 and CUDA 12.8.

The model was trained for 100 epochs at an input resolution of 640×640 with a batch size of 16. The optimizer was set to MuSGD with lr0=0.01, lrf=0.01, momentum=0.937, weight decay=5e-4, and 3 warm-up epochs. Augmentation was constrained to operations consistent with the physical realism of ground vehicles: horizontal flip (p=0.5), HSV jitter (h=0.015, s=0.4, v=0.4), and mosaic disabled for the final 10 epochs. Vertical flip and rotation were disabled, since armored vehicles do not appear upside-down or arbitrarily rotated in real surveillance imagery. A fixed seed (42) was used for reproducibility.

Training curves are shown in Fig. 3. Box and classification losses decrease smoothly on both sets, with validation closely tracking training and no sustained gap indicating overfitting. Precision plateaus near 0.94 and recall near 0.87. Both mAP@0.5 and mAP@[0.5:0.95] reach ~0.90 and ~0.63, respectively, by epoch 60 and remain stable thereafter, confirming convergence within 100 epochs.

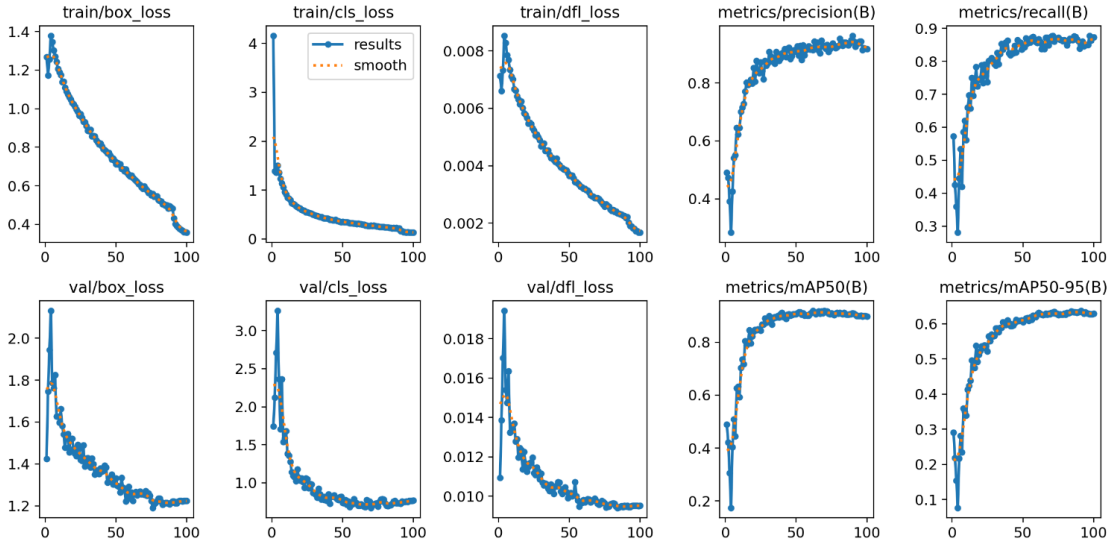


Fig. 3. Training and validation curves for YOLO26s over 100 epochs: box loss, classification loss, precision, recall, mAP@0.5, and mAP@[0.5:0.95]

The model was then evaluated on the held-out test set (304 images, 532 annotated instances). Aggregate test metrics: precision 0.937, recall 0.828, mAP@0.5 **0.896**, mAP@[0.5:0.95] **0.636**. Per-class results are summarized in Table 1.

Table 1. Per-class performance on the test set

Class	Images	Instances	Precision	Recall	mAP@0.5	mAP@[0.5:0.95]
BMP	115	167	0.939	0.834	0.927	0.675
BTR	88	149	0.932	0.831	0.873	0.648
Tank	142	216	0.940	0.819	0.887	0.585
All	304	532	0.937	0.828	0.896	0.636

BMPs achieve the highest performance across all metrics, likely reflecting their distinctive silhouette. Tanks attain comparable mAP@0.5 (0.887) but the lowest mAP@[0.5:0.95] (0.585): the model reliably finds tanks, but predicted boxes are looser at stricter IoU thresholds – a known consequence of the long-barrel geometry, where the gun barrel pushes the ground-truth box outside the visually salient region. BTRs sit between the two, with the lowest absolute mAP@0.5, reflecting their visual similarity to BMPs at small scales.

To confirm that the trained model is deployable on the target edge platform, YOLO26s was benchmarked on an NVIDIA Jetson Orin Nano Super. The PyTorch checkpoint was exported to ONNX and compiled to a TensorRT engine in FP16 precision, the standard quantization level for Jetson-class deployment. Inference was measured at an input resolution of 640×640 with a batch size of 1, averaged over the test set after a warm-up run for TensorRT kernel autotuning. The deployed engine achieved an average per-frame latency of **7.54 ms**, corresponding to a sustained throughput of **132.6 FPS** – over four times the 30 FPS threshold for real-time video processing. This leaves substantial latency headroom for downstream tracking or geolocation modules on the same device, and confirms that YOLO26s delivers competitive detection accuracy and real-time edge performance simultaneously on a low-power embedded platform.

Conclusion. This work evaluated YOLO26s for real-time military object detection on a custom dataset of approximately 12,000 images annotated for three armored vehicle classes (tank, BMP, BTR), with a particular focus on the small-object regime that dominates the dataset. The trained model achieved a test-set mAP@0.5 of 0.896 and mAP@[0.5:0.95] of 0.636, with precision 0.937 and recall 0.828. Deployed on an NVIDIA Jetson Orin Nano Super as a TensorRT FP16 engine, it sustained 132.6 FPS at 7.54 ms per frame, well above the real-time threshold. These results show that YOLO26's

NMS-free inference, DFL removal, and STAL training mechanism together deliver a detector that is both accurate and deployable on edge hardware. Future work will extend the evaluation to thermal imagery, measure power draw under sustained inference, and assess robustness in contested operational environments.

Acknowledgements. This work was realized within the framework of the program Erasmus+ Jean Monnet Module "Trustworthy artificial intelligence: the European approach" (№ 101085626 - TrustAI - ERASMUS- JMO-2022-HEI-TCH-RSCH).

References

1. Kumar, A. (2025). AUTONOMOUS DRONES: EMERGING TRENDS FROM UKRAINE. *Defence & Diplomacy*, *14*(4), 95-114.

2. Hidayatullah, P., & Tubagus, R. (2026). YOLO26: A Comprehensive Architecture Overview and Key Improvements. *arXiv preprint arXiv:2602.14582.*

3. Chakrabarty, S. YOLO26: An Analysis of NMS-Free End to End Framework for Real-Time Object Detection. arXiv 2026. *arXiv preprint arXiv:2601.12882*.

4. Zhou, M., He, S., Wang, C., & Wang, J. (2026). EFSL-YOLO: An Improved Model for Small Object Detection in UAV Vision. *Drones*, *10*(4), 243.

Сметюх А.В.
аспірант 2 курсу ІКНІ, НУ «Львівська Політехніка»
Науковий керівник к.т.н., доцент Білас О.Є., кафедра ШІ НУ «Львівська Політехніка»

ADVANCEMENTS IN LSTM-BASED TRAFFIC OPTIMIZATION: A PREDICTION MODEL

Introduction. Urban traffic congestion continues to challenge growing cities worldwide, causing economic losses and environmental degradation. Traditional prediction models such as ARIMA and SVR cannot capture the non-linear, dynamic nature of traffic patterns due to their limited capacity for modeling long-term temporal dependencies. Recent studies show that Long Short-Term Memory (LSTM) networks reduce mean absolute error by 36.7% compared to ARIMA for 60-minute predictions [1]. However, standalone LSTM models still struggle with complex spatial dependencies across road networks and integration of heterogeneous data sources such as weather and incidents [2]. Hybrid architectures combining LSTM with graph convolutional networks or attention mechanisms achieve error reductions of 29–39% [3], proving the need for further development of such models.

Problem statement. Object of research – the process of changing road traffic parameters (density, speed, occupancy) in an urban environment. Subject of research – methods and deep learning architectures based on LSTM for short- and medium-term traffic state prediction. Goal – to improve the accuracy of congestion forecasting up to 60 minutes ahead by creating a hybrid model that integrates spatial convolution, multi-scale temporal processing, and attention mechanisms, as well as to provide practical recommendations for implementing such systems. Verbal problem statement: given historical values of flow intensity, speed, density, and external factors (hour of day, day of week, weather, incidents, holidays) for a target road segment and its neighbours, predict the speed value and congestion probability at 15, 30, and 60 minutes.

Main material. The study analyses modern LSTM architectures for traffic prediction. Training on multiple correlated stations simultaneously reduces mean absolute error from 2.18 to 1.38 compared to a single station [1]. Adding external factors such as precipitation, visibility, incidents, and holidays improves prediction accuracy by 12–30% [2]. Graph convolutional networks operating with an adjacency matrix of the road network are most effective for capturing spatial dependencies between neighbouring segments. The hybrid FD-Markov-LSTM approach, which combines fundamental diagrams, Markov chains, and LSTM, achieves mean absolute error reduction of over 39% [3].

Based on these findings, we propose the MSST-LSTM (Multi-Scale Spatiotemporal LSTM) model. Its architecture consists of a graph convolutional network for spatial embedding, three parallel

LSTM branches with time windows of 5, 15, and 60 minutes, an attention mechanism to dynamically weight historical patterns, and dense layers for integrating external factors. Input parameters include traffic flow, speed, density, occupancy, station locations, an adjacency matrix, hour and day codes, holiday flag, precipitation intensity, accident severity, as well as rolling averages for 1, 3, and 24 hours. The model outputs speed forecasts, congestion probabilities, and 95% confidence intervals for horizons of 15, 30, and 60 minutes, plus a traffic state classification (free flow, moderate congestion, severe congestion).

The training process follows a three-stage strategy to ensure robust generalisation. In the first stage, each temporal branch (5, 15, 60 minutes) is pre-trained separately on its respective resolution using a mean squared error loss. The second stage fine-tunes the entire architecture jointly with a combined loss function that assigns higher weight to shorter prediction horizons. The third stage applies adversarial training using a discriminator network that distinguishes between real and predicted traffic sequences, which improves the model's ability to reproduce realistic temporal dynamics. This multi-stage approach was validated on the METR-LA dataset, showing a 5.2% additional reduction in long-horizon RMSE compared to end-to-end training from scratch.

Systematic hyperparameter optimisation was performed using a genetic algorithm with a population size of 50 over 30 generations. The optimal configuration consisted of two LSTM layers per branch with 128 hidden units each, a dropout rate of 0.3, and a learning rate of 0.001 with cosine annealing. An ablation study quantified the contribution of each architectural component: removing the spatial embedding layer increased MAE by 18%, disabling the attention mechanism increased MAE by 12%, and using only a single temporal branch (60-minute only) degraded performance by 22% on the 15-minute horizon. These results confirm that all three components are essential for achieving the target accuracy.

To meet the latency requirements of real-time traffic management systems, the model was optimised for inference on edge devices. Quantisation to 8-bit integers reduced the model size from 47 MB to 12 MB with less than 1% accuracy loss. Pruning of less significant connections (those with weights below 0.01) removed 35% of parameters while maintaining 99.2% of original performance. On a Raspberry Pi 4, inference for a single prediction step (all three horizons) takes 48 milliseconds, well within the typical 60-second update interval of traffic control systems. This makes the MSST-LSTM model suitable for deployment in resource-constrained environments such as roadside units.

Validation on METR-LA and PEMS-BAY datasets with a 70/15/15 train-validation-test split shows expected accuracy of 88–92% for the 15-minute horizon and 75–80% for the 60-minute horizon [4]. A combined loss function is used: mean absolute error for regression tasks and categorical cross-entropy for state classification. Early stopping and adaptive learning rate scheduling prevent overfitting.

Practical applications of the proposed model include dynamic route planning (18% commute time reduction in Singapore), adaptive traffic signal control (25% delay reduction in Los Angeles), and public transport optimization (36% improvement in passenger flow prediction). Data quality issues are addressed via generative adversarial networks and federated learning [1]; scalability via edge computing and model compression (40–60% latency reduction); transferability across cities via road classification features and fine-tuning on local data [5].

Conclusions. Scientific novelty – the MSST-LSTM model is the first to combine three temporal scales (5, 15, and 60 minutes) with graph spatial convolution and an attention mechanism for traffic prediction up to 60 minutes ahead. Unlike known alternatives, the proposed model provides not only point forecasts but also confidence intervals and congestion state classification, enhancing its suitability for real-time decision-making.

Practical value – specific recommendations are formulated for data preparation, architecture selection, and hyperparameter optimisation for implementing LSTM-based traffic forecasting systems. The model can be integrated into navigation applications, adaptive traffic signal control systems, and public transport dispatching platforms. An expected error reduction of 29–39% compared to traditional methods translates into an 18–25% reduction in travel delays based on pilot implementations.

Future research will focus on four main directions: first, integration of reinforcement learning for dynamic signal timing adaptation based on MSST-LSTM predictions; second, fusion of emerging

data sources such as vehicle-to-everything (V2X) communication and drone-based traffic monitoring; third, development of explainable AI methods to provide traffic operators with interpretable attention maps and feature importance scores; and fourth, deployment of the model in a federated learning framework across multiple cities without sharing sensitive traffic data. These extensions aim to create fully autonomous, privacy-preserving, and adaptive urban transportation systems.

References

1. Adedire, C. et al. (2025). *LSTM for Network Traffic Prediction*. NIPES J. Sci. Tech. Res., 7(4).

2. Kim, B.-J. & Nam, I.-W. (2025). *A Review of Hybrid LSTM Models in Smart Cities*. Processes, 13, 2298.

3. Pan, Y. A. et al. (2024). *A fundamental diagram based hybrid framework for traffic flow estimation and prediction*… Expert Syst. Appl., 238, 122219.

4. Chen, L. et al. (2025). *Temporal representation learning enhanced dynamic adversarial GCN for traffic flow prediction*. Sci Rep, 15, 8330.

5. Hadry, M. et al. (2025). *Evaluation of traffic forecasting models using traffic and context features*. Appl. Intell., 55, 755.

Загвойський Р.Ю.
аспірант кафедри АСУ НУ «Львівська політехніка»
Науковий керівник к.т.н., доцент Казимира І.Я., кафедра АСУ НУ «Львівська політехніка»

МУЛЬТИМОДАЛЬНІ МЕТОДИ AIOPS ДЛЯ УПРАВЛІННЯ ВІДМОВАМИ В ХМАРНО-ОРІЄНТОВАНИХ СИСТЕМАХ

Вступ. Масовий перехід від монолітних до мікросервісних архітектур, розгорнутих у гібридних хмарних середовищах під керуванням Kubernetes, призвів до колосального зростання обсягів телеметричних даних. За прогнозами 2024 року, до 2026 року організації з великомасштабними хмарно-орієнтованими розгортаннями оброблятимуть у середньому 248 мільярдів точок часових рядів на добу [1]. Вартість простою підприємства перевищує 23 000 доларів США на хвилину, що робить автоматизацію виявлення та діагностики відмов критично важливою. Традиційні підходи до моніторингу, що ґрунтуються на порогових значеннях метрик та ручному аналізі журналів, стають непридатними в умовах сучасної мікросервісної складності. Перспективним підходом до розв'язання цієї проблеми є інтелектуалізація IT-операцій (AIOps), яка передбачає застосування методів машинного навчання для автоматизації моніторингу, виявлення аномалій та аналізу першопричин відмов [2].

Постановка задачі. Наявні системи моніторингу та автоматизованого управління відмовами здебільшого працюють лише з одним типом телеметричних даних, не забезпечують пояснюваності рекомендацій та не здатні автоматично усувати виявлені збої. В умовах, коли мікросервісна архітектура генерує різнорідні телеметричні потоки - метрики продуктивності, журнали подій та розподілені трасування - одномодальний аналіз призводить до втрати критичної діагностичної інформації. Метою роботи є систематичний аналіз сучасних мультимодальних підходів до інтелектуалізації IT-операцій, дослідження принципів побудови архітектури MEFA-AIOps (Multimodal, Explainable, Fusion-driven AIOps), що інтегрує метрики, журнали подій та розподілені трасування, а також визначення відкритих проблем і перспектив розвитку автономних систем самовідновлення.

Основний матеріал. Сучасний конвеєр AIOps охоплює п'ять послідовних етапів: збір та нормалізацію телеметрії, виявлення відмов, аналіз першопричин, кореляцію сповіщень та автоматизовану дію. Вхідними даними слугують три канонічні модальності телеметрії: метрики продуктивності (кількісні вимірювання стану системи з інтервалами 15–60 секунд), журнали подій (напівструктуровані текстові записи про помилки та зміни стану) та розподілені трасування (записи шляху запиту через кілька сервісів із фіксацією затримок та ієрархічних залежностей). Станом на 2025 рік більшість виробничих розгортань AIOps функціонує на

другому рівні зрілості, де моделі лише надають рекомендації, а людина приймає всі рішення [3].

Архітектура MEFA-AIOps базується на чотирьох принципах: повнота модальностей (оброблення всіх трьох джерел телеметрії), достовірність злиття (комбінація модальностей стабільно перевершує найкращий одномодальний компонент), пояснюваність за задумом (кожне ранжування першопричини супроводжується каузальною атрибуцією та поясненням природною мовою) та операційна затримка менше 500 мс при швидкості 10 000 подій на секунду. Кодувальник метрик побудований на часовій згортковій мережі (TCN), кодувальник журналів використовує доопрацьовану модель RoBERTa-base із попереднім аналізом шаблонів алгоритмом Drain3, а кодувальник трасувань застосовує графову мережу з механізмом уваги (GAT) до графа залежностей сервісів. Міжмодальне злиття реалізується через механізм зваженої уваги з навчальними параметрами.

Оцінювання на наборах даних RCAEval RE2 (270 сценаріїв відмов) та RE3 (90 сценаріїв) із порівнянням 15 базових методів продемонструвало, що MEFA-AIOps досягає найвищої точності AC@1 = 0,75, що на 14 відсоткових пунктів перевищує найкращий базовий метод. Дослідження вилученням компонентів показало, що комбінація всіх трьох модальностей забезпечує покращення на 27 в. п. порівняно з найкращим одномодальним підходом (AC@1 = 0,48 для метрик), підтверджуючи комплементарну природу телеметричних джерел [4]. Контрольований експеримент із 20 черговими інженерами (SRE/DevOps, досвід ≥ 2 роки) показав, що інтерфейс пояснень MEFA-AIOps підвищив точність діагностики на 24,5 в. п. (76,8 % проти 52,3 %, $p < 0,001$), скоротив час вирішення на 48 % та отримав оцінку зручності використання SUS 81,4 (категорія «відмінно»).

Серед обмежень слід зазначити, що еталонні набори даних функціонують при навантаженнях, значно нижчих за виробничі, а одинадцять типів несправностей у RCAEval не охоплюють повний простір реальних відмов, зокрема дрейф конфігурацій та деградацію сторонніх API. Компонент генерації пояснень на основі великої мовної моделі зі стохастичними виходами залишається виробничим ризиком.

Висновки. Проведений систематичний аналіз сучасних мультимодальних підходів до AIOps засвідчив, що архітектура MEFA-AIOps із п’ятиетапним конвеєром забезпечує стабільне підвищення точності визначення першопричин відмов на 14–31 в. п. порівняно з одномодальними методами. Людинофакторний експеримент підтвердив, що пояснюваність скорочує час усунення інцидентів удвічі. Перспективними напрямками подальших досліджень є автономне усунення відмов через агентні системи, підвищення крос-середовищної узагальнюваності методами мета-навчання, федеративний моніторинг із дотриманням конфіденційності та стандартизація оцінювання якості пояснень.

Список літератури

1. Khatri P. How AI Will Reshape SAP Basis: From Monitoring to Predictive Operations. International Journal of Scientific Research in Engineering and Management, 2025, vol. 09, no. 06. DOI: 10.55041/IJSREM51046.

2. Notaro P., Cardoso J., Gerndt M. A Survey of AIOps Methods for Failure Management. ACM Transactions on Intelligent Systems and Technology, 2021, vol. 12, no. 6, pp. 1–45. DOI: 10.1145/3483424.

3. Chen Z. et al. A Survey of AIOps in the Era of Large Language Models. ACM Computing Surveys, 2025. DOI: 10.1145/3746635.

4. Pham L., Zhang H., Ha H., Salim F., Zhang X. RCAEval: A Benchmark for Root Cause Analysis of Microservice Systems with Telemetry Data. Companion Proceedings of the ACM Web Conference 2025 (WWW’25), pp. 777–780. DOI: 10.1145/3701716.3715290.

5. Liu Y. et al. MAD-CMC: Unsupervised Microservice System Anomaly Detection via Contrastive Multi-Modal Representation Clustering. Information Processing & Management, 2024. DOI: 10.1016/j.ipm.2024.103728.

Фаєрчук В. В.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

МОДУЛЬНИЙ ФРЕЙМВОРК ДЛЯ ВИЯВЛЕННЯ AI-ЗГЕНЕРОВАНИХ ЗОБРАЖЕНЬ ОБЛИЧ НА ОСНОВІ ГІБРИДНОЇ АРХІТЕКТУРИ CNN+GNN

Вступ. У сучасному цифровому середовищі зображення та відео з людськими обличчями стали одним із основних способів комунікації. Соціальні мережі, відеоплатформи та новинні ресурси щоденно поширюють величезну кількість візуального контенту, проте розвиток технологій штучного інтелекту, зокрема StyleGAN2, Stable Diffusion та DALL-E, призвів до появи фотореалістичних синтетичних зображень, які практично неможливо відрізнити від справжніх. Це створює серйозні загрози в галузі кібербезпеки та медіа-верифікації, оскільки обсяг deepfake-контенту зростає щорічно [1, 2]. Актуальність дослідження зумовлена потребою у надійних автоматизованих системах, здатних виявляти ознаки синтетичного походження у складних структурних деталях обличчя.

Постановка задачі. Об'єкт дослідження – процес автоматизованого виявлення AI-згенерованих зображень облич людей. Предмет дослідження – засоби проектування модульного фреймворку на основі гібридної CNN+GNN архітектури. Мета дослідження – розробити модульний фреймворк FaceAI-Pipeline для автоматизованого виявлення AI-згенерованих зображень облич з використанням методів глибокого навчання.

Основний матеріал. Фреймворк FaceAI-Pipeline спроєктований за принципом модульної архітектури з чітким розділенням відповідальностей. Система складається з таких модулів: завантаження даних (Data Module), попередньої обробки (Preprocessing Module), навчання (Training Module), оцінки метрик (Evaluation Module) та сервісного модуля (Serving Module). Кожен із цих компонентів забезпечує послідовний обчислювальний процес, що відповідає сучасним принципам програмної інженерії.

Архітектура програмної системи базується на використанні гібридної нейронної мережі. Загальна архітектура фреймворку наведена на рисунку 1, а детальна схема гібридної мережі із зазначенням розмірностей - на рисунку 2.

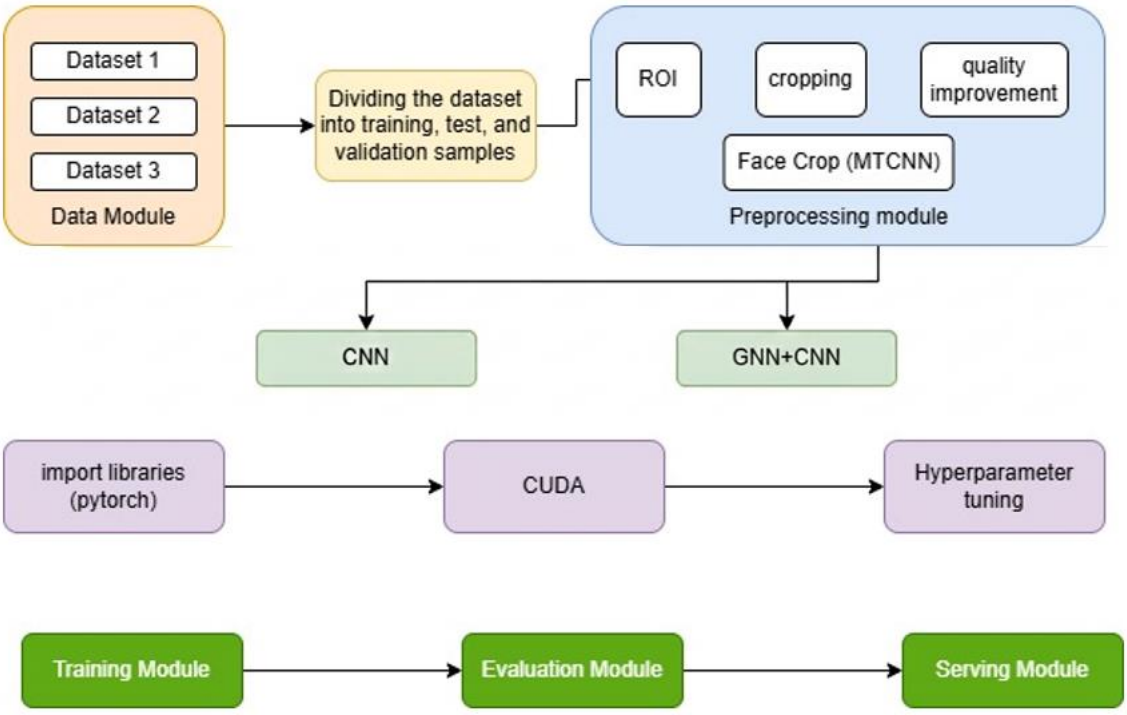


Рисунок 1 – Загальна архітектура фреймворку FaceAI-Pipeline

Рисунок 2 – Архітектура гібридної нейронної мережі CNN+GNN

Ключовим компонентом є модуль попередньої обробки, який використовує детектор MTCNN [3] для виявлення та нормалізації області обличчя. Це дозволяє усунути проблему dataset bias, коли модель навчається на фонових артефактах конкретного набору даних замість реальних ознак AI-генерації. Система розроблена на мові Python із використанням бібліотек PyTorch та OpenCV.

Гібридна архітектура поєднує згорткову мережу EfficientNet-B2 [4] як екстрактор локальних ознак та графову мережу Graph Attention Network (GAT) [5]. Зображення розбивається на патчі 16×16 пікселів, що формують 256 вузлів графу із сітковою топологією. Механізм multi-head attention у GAT-модулі дозволяє навчати зважені залежності між фрагментами зображення. Порівняння характеристик архітектур CNN Baseline та CNN+GNN наведено у таблиці 1.

Таблиця 1 – Порівняння архітектур CNN Baseline та CNN+GNN

Характеристика	CNN Baseline	CNN+GNN
Backbone	EfficientNet-B2	EfficientNet-B2
Розмір вектора ознак	1 408	1 920 (1408+256+256)
Додатковий модуль	-	GATModule (3 шари, 4 головки)
Кількість параметрів	~0.7M (trainable)	~5M (trainable)
Час inference (GPU GTX 1650)	0.18–0.28 с	0.35–0.50 с
Рекомендований датасет	> 1 000 зобр.	> 5 000 зобр.

Для навчання використано реальні фото з датасету FFHQ [6] та синтетичні зображення від компанії Generated Photos [7]. Експерименти проводилися на локальній станції з GPU NVIDIA GeForce GTX 1650. Характеристики використаних датасетів відображені у таблиці 2.

Таблиця 2 – Характеристики використаних датасетів

Характеристика	FFHQ (Real)	generated.photos (AI)
Загальний обсяг	70 000 зображень	100 000+ зображень
Використано у роботі	10 000	10 000
Оригінальна роздільність	1024×1024 px	512×512 px
Після MTCNN кропу	256×256 px	256×256 px
Генератор	Flickr (реальні фото)	StyleGAN2
Розбиття train/val/test	80 / 10 / 10 %	80 / 10 / 10 %

Результати тестування на вибірці з 2 000 зображень показали, що гібридна архітектура перевершує CNN Baseline за всіма метриками, досягаючи Accuracy 95.31%. Криві навчання та ROC-криві для обох моделей зображені на рисунках 3 та 4 відповідно.

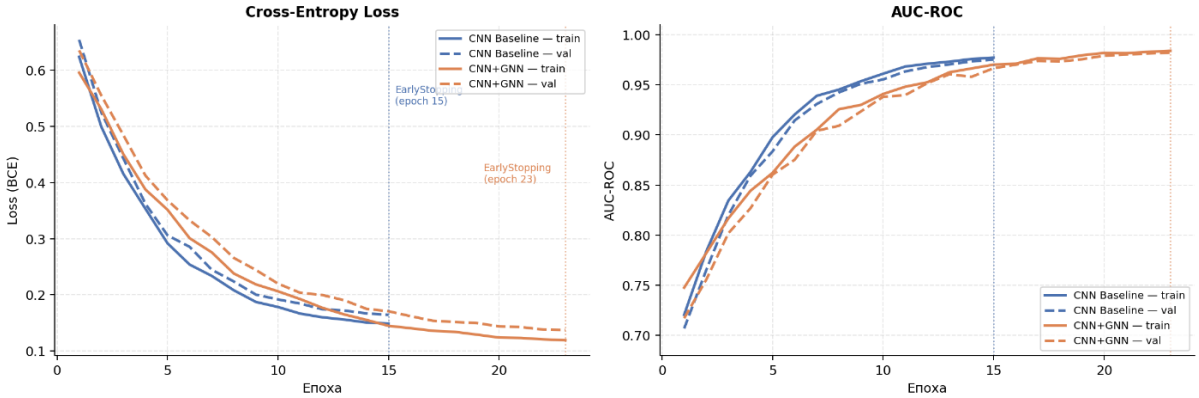


Рисунок 3 – Криві навчання CNN Baseline та CNN+GNN

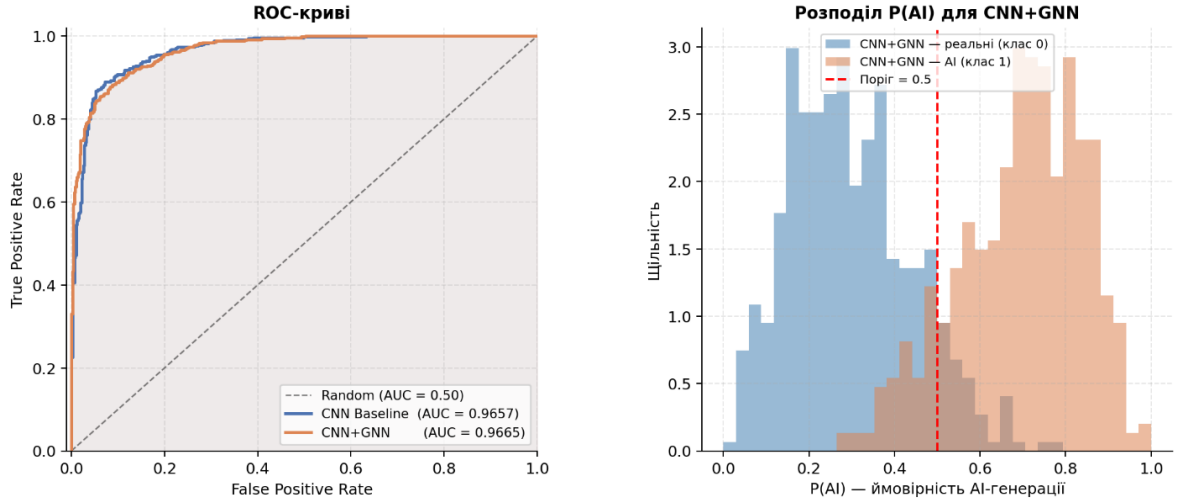


Рисунок 4 – ROC-криві та розподіл імовірностей P(AI)

Важливою перевагою CNN+GNN є вищий показник Recall, що мінімізує ризики пропуску deepfake-контенту. Час обробки одного зображення через Serving Module складає в середньому 0.42 с для гібридної моделі.

Висновки. У рамках роботи створено програмний фреймворк FaceAI-Pipeline, який забезпечує автоматичну детекцію та класифікацію синтетичних зображень. Виявлено та усунуто проблему системного зміщення даних шляхом нормалізації області обличчя. Гібридна архітектура підтвердила свою ефективність у моделюванні просторових залежностей між патчами, що дозволило підвищити точність діагностування підробок. Практичне застосування системи є важливим для медіа-верифікації та захисту систем біометричної ідентифікації.

Список літератури

1. Gong L.Y., Li X.J. A contemporary survey on deepfake detection: Datasets, algorithms, and challenges // Electronics. 2024. Vol. 13, No. 3. P. 585.

2. Zhao H. et al. Multi-attentional Deepfake Detection. Proc. CVPR. 2021. P. 2185–2194.

3. Transcending Forgery Specificity with Latent Space Augmentation for Generalizable Deepfake Detection (CVPR 2024).

4. Vivekananda G.N. et al. Refining digital security with EfficientNetV2-B2 deepfake detection (2025).

5. She H. et al. Using Graph Neural Networks to Improve Generalization Capability of the Models for Deepfake Detection (2024).

6. FFHQ: Flickr-Faces-HQ Dataset. URL: https://github.com/NVlabs/ffhq-dataset.

7. Generated Photos. 100K Faces Dataset. URL: https://generated.photos.

Басалкевич О.А.
аспірант 3 курсу ІКНІ НУ Львівська Політехніка
Науковий керівник д.т.н., ст.н.с. Рудавський Д.В., кафедра АСУ НУ Львівська Політехніка

КОНЦЕПЦІЯ БАЗОВОЇ МУЛЬТИАГЕНТНОЇ АРХІТЕКТУРИ ДЛЯ ОЦІНЮВАННЯ ВТОМНОГО РЕСУРСУ КОРЕНІВ ЛОПАТЕЙ ВІТРОГЕНЕРАТОРІВ

Вступ. Тенденція до збільшення габаритів сучасних вітрогенераторів зумовлює зростання циклічних механічних навантажень на корені лопатей. Точне прогнозування їхньої втомної довговічності є критично важливим для забезпечення надійності конструкцій [1]. Отримання достовірних оцінок вимагає застосування комплексних математичних моделей, які враховують стохастичну природу вітру та процеси накопичення пошкоджень, а також проведення серії ресурсомістких комп'ютерних симуляцій [2]. Через це задача оцінки втомного ресурсу виходить за межі суто прикладної механіки, перетворюючись на складну проблему інженерії програмного забезпечення та високопродуктивних обчислень.

Постановка задачі. Об'єкт дослідження – процеси високопродуктивних обчислень під час комп'ютерного моделювання динаміки вітрових турбін. Предмет дослідження – програмна архітектура та методи організації паралельної взаємодії обчислювальних агентів для системи оцінки втомного ресурсу коренів лопатей вітрогенератора. Головна мета даного дослідження полягає в розробленні та обґрунтуванні концепції базової архітектури для мультиагентної системи прогнозування втомної довговічності коренів лопатей вітрових турбін із використанням паралельних розрахунків. Запропонована модель дозволить оптимізувати розподіл обчислювального навантаження та скоротити загальний час моделювання без втрати точності фінальних результатів.

Основний матеріал. В основу розробленої системи покладено мультиагентний підхід [3], що реалізується через асинхронну взаємодію функціональних модулів: парсера вхідних даних, агентів TurbSim та OpenFAST, модуля аналізу втоми (Fatigue Analyser) та контролера логів (Logs Checker).

Функціональну основу архітектури становить ієрархія класів, побудована на засадах успадкування та композиції (див. рис. 1). Її базовим компонентом є абстрактний клас AgentBase, який регламентує єдиний життєвий цикл для всіх розрахункових модулів. Він інкапсулює механізми розпаралелювання завдань за допомогою пулу потоків (ThreadPoolExecutor), де кількість активних потоків адаптується під апаратні можливості обчислювальної системи. Для надійного управління життєвим циклом сторонніх розрахункових процесів (TurbSim, OpenFAST) розроблено клас ProcessesManager, який використовує патерн «Snapshot» для безпечного примусового термінування потоків без ризику взаємних блокувань. Глобальна синхронізація агентів та миттєва реакція на аварійні зупинки реалізовані через механізм міжпроцесної розділюваної пам'яті (SharedMemory).

Для організації безперервного обчислювального конвеєра спроєктовано спеціалізований клас FolderObservingAgent, який є нащадком AgentBase. Він впроваджує подієво-орієнтовану модель взаємодії. Завдяки інтеграції з бібліотекою watchdog та спеціалізованому обробнику AsyncAppearedFileHandler, агенти працюють у режимі активного моніторингу визначених директорій файлової системи. Поява нового вихідного файлу від попереднього модуля автоматично фіксується системою подій, проходить валідацію за маскою та миттєво стає тригером для асинхронного запуску наступного етапу обчислень у вільному робочому потоці.

Перевагою запропонованої моделі є формування ефективного конвеєра обробки даних: TurbSim-агент паралельно генерує випадкові поля швидкостей вітру; на їхній основі OpenFAST-агент одразу ініціює розрахунок аеропружної динаміки турбіни; після чого модуль Fatigue Analyser у міру надходження out-файлів асинхронно обчислює та акумулює втомні пошкодження для кожного сектора лопаті. Паралельно Logs Checker аналізує журнали симуляцій і, у разі виявлення критичних помилок, здатний миттєво зупинити всі процеси системи через зміну сигнального байта у розділюваній пам'яті.

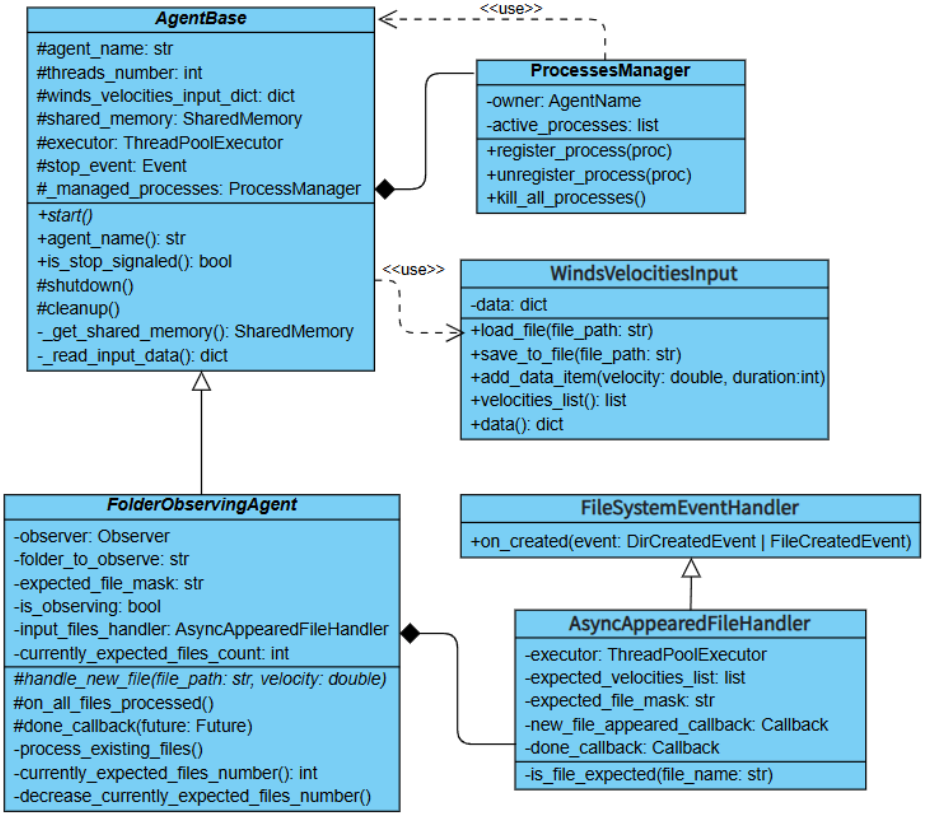


Рисунок 1 – UML-діаграма класів спільної архітектури модулів TurbSim Module, OpenFAST Module, Fatigue Analyser Module та Logs Checker

Запропонований підхід усуває простої обчислювальних ресурсів, характерні для традиційного послідовного виконання, оптимізує навантаження та робить систему легко масштабованою для задач будь-якої обчислювальної складності.

Висновки. Запропонована мультиагентна архітектура забезпечує ефективне розпаралелювання та безперервну конвеєрну обробку даних у задачах оцінювання втомного ресурсу лопатей вітрових турбін. Завдяки подієво-орієнтованій взаємодії незалежних програмних агентів усуваються простої обчислювальних ресурсів, що дозволяє суттєво скоротити загальний час складного комп'ютерного моделювання. Загалом, такий підхід створює надійне підґрунтя для подальшого масштабування обчислювальної системи та підвищення деталізації прогнозів без критичного зростання часових витрат.

Список літератури

1. Wang, W., Xue, Y., He, C., Zhao, Y. (2022). Review of the Typical Damage and Damage-Detection Methods of Large Wind Turbine Blades. Energies 2022, 15(15), 5672. https://doi.org/10.3390/en15155672

2. Basalkevych, O. A., Rudavskyi D. V. (2025). Model and software module for calculating fatigue damage accumulation in the wind turbine blade root. Ukrainian Journal of Information Technologies, 2025, 7(1), 45-51. https://doi.org/10.23939/ujit2025.01.045

3. Basalkevych, O. A., Rudavskyi D. V. (2025). Virtual scientific laboratory «Wind tunnel» for investigating the fatigue durability of wind turbine blade roots. Journal of Lviv Polytechnic National University "Information Systems and Networks", 2025, 18(1), 26-34. https://doi.org/10.23939/sisn2025.18.1.026 [in Ukrainian].

Вовк В.С.
студент 4 курсу ФКІТ, Західноукраїнський національний університет
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ, Західноукраїнський національний університет

ГЕНЕРУВАННЯ ХУДОЖНІХ ЗОБРАЖЕНЬ ЗА ТЕКСТОВИМ ОПИСОМ

Вступ. Зростання попиту на цифровий візуальний контент зумовлює широке застосування систем генерації зображень за текстовим описом (text-to-image, T2I). Сучасні дифузійні моделі здатні синтезувати зображення студійної якості з простого природньомовного запиту, проте їх практичне використання потребує вибору між двома принципово різними режимами: локальним виконанням на GPU користувача та хмарними API. Локальний інференс гарантує повну приватність даних, тоді як хмарні сервіси забезпечують доступність без потужного обладнання. Актуальність роботи зумовлена відсутністю зручного настільного інструменту під Windows, який поєднував би обидва підходи в єдиному конвеєрі з локальним журналюванням результатів.

Постановка задачі. Об'єкт дослідження – процес синтезу зображень за текстовим описом у настільному застосунку. Предмет дослідження – програмні методи реалізації T2I-пайплайнів, обробки промптів та збереження історії генерацій. Мета дослідження – розробити GUI-застосунок на базі CustomTkinter з підтримкою локального виконання через бібліотеку Diffusers та хмарних API (Gemini, Stability AI), використанням SQLite для зберігання шаблонів і журналу генерацій, а також гнучкою конфігурацією через файл config.env.

Основний матеріал. Перед проектуванням застосунку проведено порівняльний аналіз локального та хмарного режимів виконання. Локальний режим забезпечує повний контроль над даними і не залежить від підключення до мережі, проте потребує наявності відеокарти з достатнім обсягом VRAM. Хмарний режим знижує вимоги до апаратури, але пов'язаний з передачею промптів і зображень до сторонніх серверів та тарифікацією запитів. Результати порівняльного аналізу узагальнено в таблиці 1.

Таблиця 1 – Порівняльний аналіз локального та хмарного режимів генерації

Критерій	Локально (Diffusers)	Хмарний API
Обчислення	GPU бажана; CPU можливий, але повільний	На стороні провайдера
Приватність	Дані залишаються на ПК користувача	Передача згідно з умовами сервісу
Вартість	Апаратура та електроенергія	Підписка або оплата за запити
Доступні моделі	Hugging Face Hub	Каталог API, агрегатори

Застосунок реалізовано мовою Python 3.10+ з використанням бібліотек PyTorch, Diffusers, Transformers та Pillow. Для забезпечення плавності інтерфейсу всі тривалі операції (завантаження моделі, генерація) виконуються у фонових потоках, відокремлених від головного потоку Tkinter. Промпт нормалізується, за потреби доповнюється шаблонами з бази даних або розширюється за допомогою LLM; для моделі Stable Diffusion 1.5 додатково контролюється ліміт у 77 токенів CLIP-кодувальника.

Загальну схему інференсу зображено на рисунку 1. Процес розпочинається з ініціалізації системи та прийому вхідних даних: промпту, негативного промпту й параметрів інференсу. Після кодування тексту й умовних ознак система визначає режим виконання – локальний або хмарний – та ініціалізує відповідний бекенд. Далі виконується зведений запит із шумом z_T, ітеративний денойзинг і декодування латентного вектора z_0 у пікселі засобами VAE-декодера. Завершується процес нормалізацією та збереженням файлу.

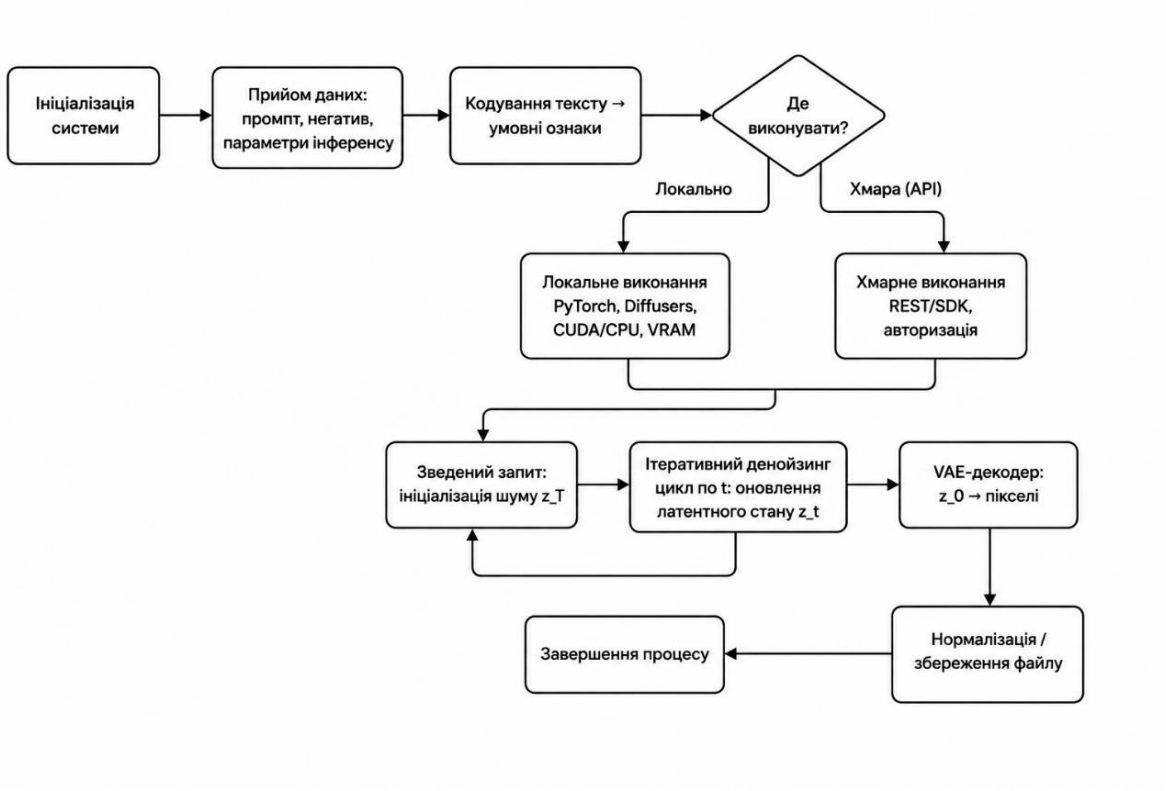


Рисунок 1 – Узагальнена схема інференсу text-to-image

Персистентність даних забезпечує вбудована СУБД SQLite, що не потребує окремого серверного процесу. База даних зберігає повний журнал генерацій, пресети параметрів, шаблони промптів, теги та шляхи до збережених файлів. Перелік основних таблиць наведено в таблиці 2.

Таблиця 2 – Основні таблиці бази даних SQLite

Таблиця	Призначення
models	Бекенди та ключі підключення до моделей
generation_presets	Збережені набори параметрів інференсу
prompt_templates	Шаблони для автоматичного розширення промптів
generation_history	Журнал генерацій із шляхами до файлів
tags, generation_tags	Мітки та їх зв'язок із записами журналу
image_variants	Повне зображення та мініатюра прев'ю

ER-модель бази даних наведено на рисунку 2. Центральною таблицею є generation_history, яка пов'язана з усіма іншими сутностями: вона посилається на модель (models), пресет параметрів (generation_presets) та шаблон промпту (prompt_templates). Таблиця image_variants зберігає повне зображення та мініатюру, пов'язані з конкретним записом журналу. Зв'язок «багато до багатьох» між генераціями та мітками реалізовано через проміжну таблицю generation_tags.

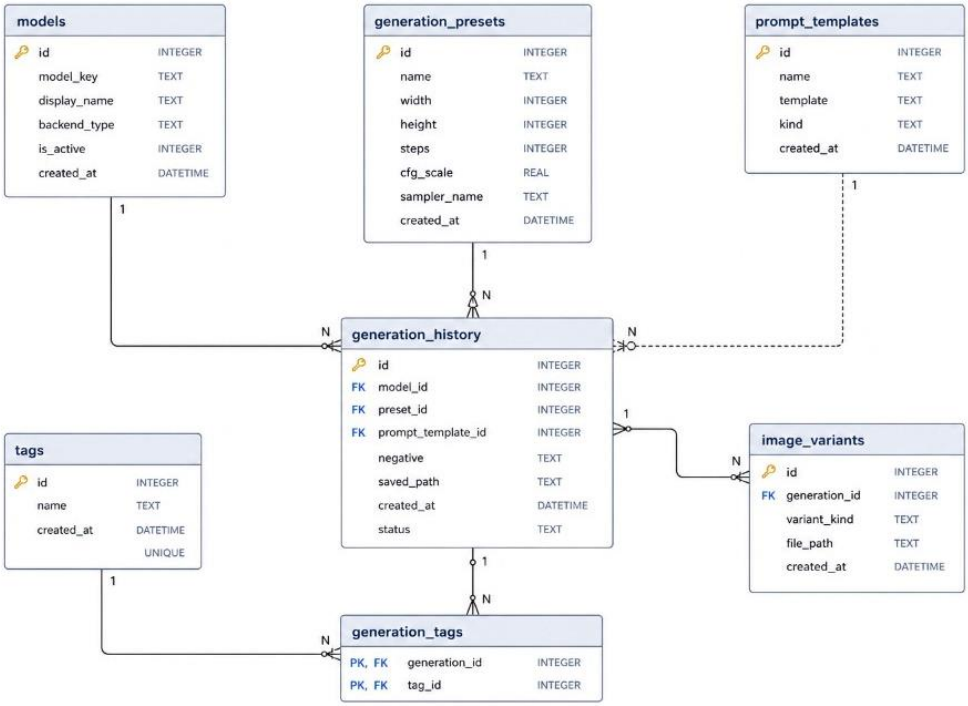


Рисунок 2 – ER-модель бази даних SQLite

Ключовим елементом застосунку є головне вікно, що надає доступ до всіх функцій генерації. Панель зліва містить: вибір моделі (локальна або API), поля введення промпта та негативного промпта, вибір шаблону типу зображення, кнопки поліпшення промпта через LLM, параметри кількості кроків дифузії, значення CFG-шкали та співвідношення сторін. Після натискання кнопки «Згенерувати» результат відображається у правій зоні прев'ю, а запис автоматично зберігається до журналу SQLite. Інтерфейс головного вікна наведено на рисунку 3.

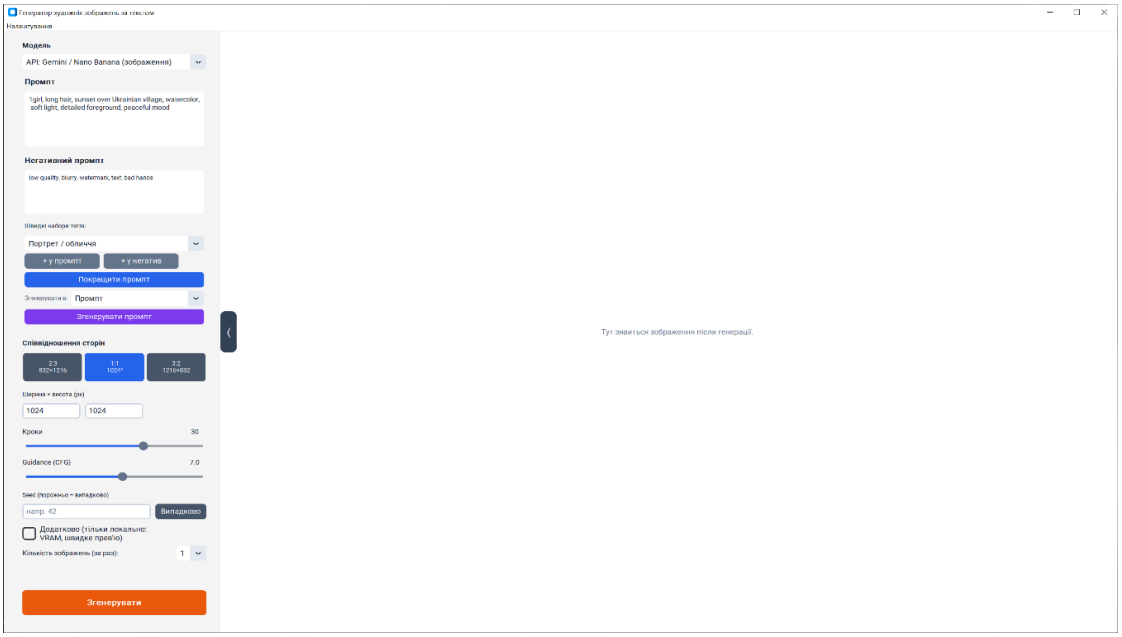


Рисунок 3 – Головне вікно застосунку «Генератор художніх зображень за текстом»

Висновки. У роботі розроблено настільний застосунок для генерації зображень за текстовим описом з підтримкою локального виконання через Diffusers та хмарних API. Реалізовано конвеєр підготовки промптів із нормалізацією, шаблонами та контролем токенів CLIP, а також SQLite-журнал для збереження повної історії генерацій. Проведено порівняльний аналіз локального й хмарного режимів, описано схему інференсу та модель даних. Перспективи

подальшого розвитку: підтримка черги завдань, інтеграція ControlNet та LoRA-адаптерів, впровадження розширених метрик оцінки якості зображень.

Список літератури

1. Rombach R. et al. High-Resolution Image Synthesis with Latent Diffusion Models // CVPR. – 2022.
2. Hugging Face. Diffusers documentation. URL: https://huggingface.co/docs/diffusers (дата звернення: 19.05.2026).
3. Radford A. et al. Learning Transferable Visual Models from Natural Language Supervision // arXiv:2103.00020. – 2021.
4. CustomTkinter. Documentation. URL: https://customtkinter.tomschimansky.com (дата звернення: 19.05.2026).
5. SQLite Documentation. URL: https://www.sqlite.org/docs.html (дата звернення: 19.05.2026).
6. Google AI for Developers. Gemini API. URL: https://ai.google.dev (дата звернення: 19.05.2026).

Нагина М. В.[1], Зварич В. І.[2]
[1]студент 4 курсу АСУ НУЛП, [2]асистент каф. АСУ НУЛП
Науковий керівник к.т.н., асистент Зварич В. І., кафедра АСУ НУЛП

ПОЗИТИВНО-КЛАСОВИЙ ПІДХІД ДО ВИЯВЛЕННЯ ПОРУШЕНЬ НОСІННЯ ЗАСОБІВ ІНДИВІДУАЛЬНОГО ЗАХИСТУ

Вступ. На промислових і будівельних об'єктах значна частка нещасних випадків пов'язана з тим, що працівники не використовують засоби індивідуального захисту (ЗІЗ): каски, сигнальні жилети, спецвзуття. Ручний інспекційний контроль обмежений присутністю інспектора в одному місці, тому задача автоматизації цього контролю засобами комп'ютерного зору є практично затребуваною. Поява одноетапних детекторів сімейства YOLO [1, 2] зробила це технічно можливим у реальному часі. Водночас прямий шлях – тренування моделі на класах-відсутностях (no_helmet, no_vest) - на публічних датасетах дає повноту лише 0.33-0.40 [3] через неоднозначну розмітку «відсутності об'єкта».

Постановка задачі. Об'єктом дослідження є процеси автоматизованого контролю використання ЗІЗ, предметом - алгоритми виявлення відповідних порушень за відеоданими. Мета роботи – побудувати алгоритм виявлення порушень носіння ЗІЗ, який обходить характерну для прямої 8-класової схеми неоднозначність розмітки.

Основний матеріал. Запропоновано двоетапну схему. Перший етап – детектор D, який виконує інференс YOLO11s на повному кадрі та повертає множину детекцій лише позитивних класів C = {helmet, vest, person}; модель не «знає» про порушення. Другий етап – аналізатор A, який працює над результатами детектора. Для кожної рамки людини p визначається зона голови H(p) (верхня третина її рамки) та зона торса T(p) (середня третина). Порушення no_helmet фіксується, коли жодна рамка каски не задовольняє умову асоціації:

$$\forall h \in \{d : c(d) = helmet\} : \; IoU(b(h), H(p)) < \tau assoc,$$

де $IoU(b_1, b_2) = |b_1 \cap b_2| / |b_1 \cup b_2|$ - коефіцієнт перетину рамок до об'єднання, $\tau\_assoc = 0.1$. Для no_vest умова симетрична, але на зону торса.

На будмайданчиках працівники часто носять куртки сигнальних кольорів без сертифікованого жилета. Формально це no_vest, але фактично такий працівник захищений у тому сенсі, який ставила собі задача охорони праці – його видно. Тому додано виняток за колірною часткою у зоні торса, обчисленою у просторі HSV (стабільніший за RGB до зміни освітленості). Задано шість діапазонів сигнальних кольорів. Якщо сумарна частка пікселів торса у цих діапазонах перевищує 0.20 – порушення no_vest не записується.

Узагальнену схему двоетапного підходу наведено на рисунку 1.

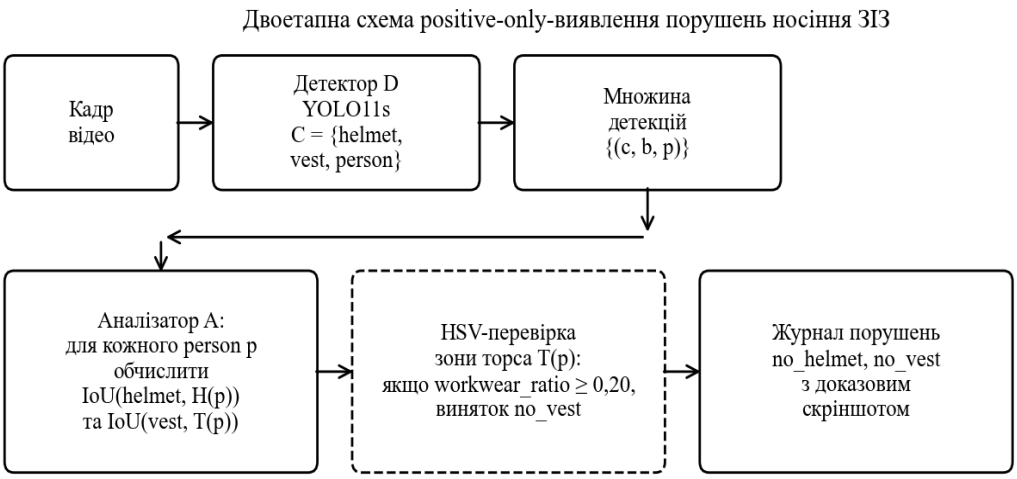


Рисунок 1 – Двоетапна схема positive-only-виявлення порушень носіння ЗІЗ

Підхід перевірено на курованому датасеті positive_core (близько 9 тисяч зображень з трьох публічних джерел: Hard Hat Workers, Construction Site Safety, Color Helmet and Vest). Як базова архітектура взято YOLO11s з 9,4 млн параметрів; тренування виконано за стратегією transfer learning з вагами COCO. На тестовому наборі отримано mAP@0.5 = 0.810 для трикласової схеми та 0.831 для шестикласової (з додатковими класами окулярів, рукавиць, спецвзуття). Середня латентність інференсу одного зображення на GPU з 8 ГБ VRAM становить 37-44 мс, що задовольняє вимоги режиму реального часу. Приклад роботи системи на реальному кадрі наведено на рисунку 2.

Рисунок 2 – Приклад роботи системи: коректне виявлення позитивних класів (helmet, vest, person) для працівника у спорядженні та фіксація порушення no_helmet для особи без захисної каски

Висновки. Запропонований позитивно-класовий підхід замінює неоднозначну класифікацію класів-відсутностей логічним висновком на основі геометричного аналізу позитивних детекцій, а додатковий етап HSV-сегментації коректно опрацьовує пограничний випадок з робочим одягом. У результаті отримано mAP@0.5 = 0.810-0.831 при латентності 37-44 мс на зображення, тоді як прямий 8-класовий підхід на тих самих даних дає повноту 0.33-0.40 [3]. Отримані результати свідчать про потенційну придатність підходу для подальшого тестування в умовах відеоспостереження на промислових об'єктах.

Список літератури

1. Khanam R., Hussain M. YOLOv11: An Overview of the Key Architectural Enhancements. arXiv preprint. 2024. DOI: 10.48550/arXiv.2410.17725.

2. Ultralytics YOLO11: official documentation. URL: https://docs.ultralytics.com/models/yolo11/ (дата звернення: 09.05.2026).

3. Ahmed M. I. B., Saraireh L., Rahman A. et al. Personal Protective Equipment Detection: A Deep-Learning-Based Sustainable Approach. *Sustainability*. **2023**. Vol. 15, no. 18. 13990. DOI: 10.3390/su151813990.

4. Akyon F. C., Altinuc S. O., Temizel A. Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection. *2022 IEEE International Conference on Image Processing* (ICIP). **2022**. P. 966–970. DOI: 10.1109/ICIP46576.2022.9897990.

5. Liu, Y., Jiang, B., He, H., Chen, Z., Xu, Z. Helmet wearing detection algorithm based on improved YOLOv5. *Scientific Reports*. **2024**. Vol. 14. 8768. DOI: 10.1038/s41598-024-58800-6.

Бойко Н.Р.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ГІБРИДНИХ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ

Вступ. У сучасних умовах стрімкого розвитку інформаційних технологій та цифровізації практично всіх сфер суспільного життя питання інформаційної безпеки набувають особливої актуальності. Однією з найбільш поширених загроз для комп'ютерних систем і мереж є шкідливе програмне забезпечення, яке застосовується для несанкціонованого доступу до інформаційних ресурсів, порушення цілісності та конфіденційності даних, а також дестабілізації роботи інформаційних систем. Зростання кількості та різноманітності зразків шкідливого програмного забезпечення, зокрема поліморфних і метаморфних програм, суттєво ускладнює використання традиційних сигнатурних методів виявлення [1, 2].

Постановка задачі. Проведений аналіз предметної області та існуючих методів виявлення шкідливого програмного забезпечення показав, що традиційні підходи не забезпечують необхідного рівня ефективності в умовах постійного ускладнення архітектури сучасного шкідливого програмного забезпечення та зростання кількості нових і модифікованих загроз. Зокрема, сигнатурні методи виявлення є малоефективними щодо невідомих зразків, тоді як поведінкові та евристичні підходи характеризуються високою ресурсоємністю та складністю практичної реалізації. Методи глибокого навчання, попри їх високу точність, часто потребують значних обчислювальних ресурсів і не завжди адаптовані до використання у вигляді прикладних програмних модулів [3, 4].

У зв'язку з цим актуальним є завдання розроблення програмного модуля для виявлення шкідливого програмного забезпечення, який поєднує переваги гібридних моделей глибокого навчання з можливістю практичного використання в системах інформаційної безпеки. Такий модуль повинен забезпечувати високу точність класифікації, стійкість до методів обфускації коду, а також прийнятні вимоги до обчислювальних ресурсів.

Об'єктом дослідження є процеси виявлення шкідливого програмного забезпечення в комп'ютерних системах. Предметом дослідження є методи та алгоритми виявлення шкідливого програмного забезпечення на основі гібридних моделей глибокого навчання. Метою роботи є розроблення програмного модуля виявлення шкідливого програмного забезпечення на основі гібридних моделей глибокого навчання, який забезпечує підвищення ефективності класифікації шкідливих програм за рахунок поєднання ознак, отриманих різними нейронними архітектурами.

Основний матеріал. Архітектура програмного модуля виявлення шкідливого програмного забезпечення включає такі основні компоненти (рисунок 1):

- завантаження та керування вхідними даними;
- попередня обробка даних;
- гібридна модель глибокого навчання;
- класифікація та прийняття рішень;
- формування та збереження результатів.

Кожен із зазначених компонентів виконує чітко визначені функції та взаємодіє з іншими компонентами через стандартизовані інтерфейси, що забезпечує незалежність реалізації окремих модулів і можливість їх заміни або модернізації без суттєвих змін у загальній структурі системи.

Етап завантаження вхідних даних відповідає за приймання виконуваних файлів або їхніх числових представлень, які надходять на аналіз. У межах даного етапу здійснюється перевірка коректності вхідних даних, визначення формату файлів та організація їх подальшої обробки. Модуль повинен підтримувати як одиничний аналіз файлів, так і пакетну обробку множини зразків, що є важливим для експериментального дослідження та практичного застосування програмного модуля.

Рисунок 1 – Структурна схема програмного модуля виявлення шкідливого програмного забезпечення

З метою підвищення гнучкості архітектури етап завантаження реалізується незалежно від конкретного способу подання даних, що дозволяє у подальшому розширити перелік підтримуваних форматів без зміни логіки інших компонентів системи.

Етап попередньої обробки даних призначений для перетворення вхідних зразків програмного забезпечення у формат, придатний для подання на вхід гібридної моделі глибокого навчання. Залежно від обраного підходу, на даному етапі може виконуватися зчитування байтового вмісту файлів, нормалізація даних, формування двовимірних матриць або зображень, а також приведення даних до фіксованого розміру.

Попередня обробка відіграє важливу роль у забезпеченні стабільності та точності роботи моделей глибокого навчання, оскільки якість вхідних даних безпосередньо впливає на результати класифікації. Тому етап попередньої обробки реалізується з можливістю налаштування параметрів, що дозволяє адаптувати його до різних наборів даних і архітектур нейронних мереж.

Ключовим компонентом архітектури програмного модуля є гібридна модель глибокого навчання. Вона відповідає за безпосередній аналіз підготовлених даних і формування ознакових представлень, що використовуються для класифікації програмного забезпечення. Гібридна модель поєднує декілька згорткових нейронних архітектур, що дозволяє використовувати їхні сильні сторони та підвищити загальну ефективність виявлення.

У межах даного етапу реалізується процес завантаження попередньо навчених моделей, їх паралельне або послідовне застосування до вхідних даних, а також об’єднання отриманих ознакових векторів. Такий підхід забезпечує більш повне представлення характеристик аналізованих зразків програмного забезпечення та підвищує стійкість до методів обфускації коду.

Етап класифікації здійснює аналіз ознакових представлень, сформованих гібридною моделлю глибокого навчання, та визначає належність аналізованого зразка до певного класу програмного забезпечення. На даному етапі може використовуватися окремий класифікатор або вихідні шари нейронних мереж, які формують імовірнісні оцінки для кожного класу.

Результатом роботи даного етапу є рішення щодо класифікації зразка, а також числова оцінка рівня впевненості, що дозволяє використовувати результати аналізу для подальшої автоматизованої обробки або прийняття рішень у складі комплексних систем захисту інформації.

Етап формування результатів відповідає за представлення вихідної інформації у зручному для користувача або зовнішніх систем форматі. До функцій даного етапу належить виведення результатів класифікації, збереження їх у файлах або базах даних, а також формування звітів про роботу програмного модуля.

Наявність окремого етапу для обробки результатів забезпечує можливість інтеграції програмного модуля в більші інформаційні системи та спрощує аналіз ефективності його роботи під час експериментальних досліджень.

Взаємодія між компонентами програмного модуля реалізується за принципом послідовної обробки даних, що відповідає логіці процесу виявлення шкідливого програмного забезпечення. Така архітектура дозволяє чітко відокремити етапи обробки та забезпечує прозорість

функціонування модуля.

Запропонована архітектура є гнучкою та масштабованою, що дозволяє адаптувати програмний модуль до різних умов експлуатації, змінювати склад гібридної моделі глибокого навчання та розширювати функціональність без суттєвих змін у базовій структурі.

Враховуючи архітектуру програмного модуля, наведену на рисунку 1, алгоритм його роботи реалізується у вигляді послідовної обробки вхідних даних. Такий підхід відповідає логіці задачі виявлення шкідливого програмного забезпечення, дозволяє чітко розмежувати етапи обробки та забезпечує прозорість і відтворюваність результатів.

Процес роботи програмного модуля починається з отримання вхідних даних і завершується формуванням та збереженням результатів класифікації. На кожному етапі алгоритму виконуються чітко визначені операції, які забезпечують підготовку даних, їх аналіз за допомогою гібридної моделі глибокого навчання та інтерпретацію отриманих результатів.

Алгоритм роботи програмного модуля включає такі основні етапи:

- приймання та перевірка вхідних даних;
- попередня обробка виконуваних файлів;
- формування ознакових представлень;
- аналіз за допомогою гібридної моделі глибокого навчання;
- класифікація та прийняття рішення;
- формування, збереження та/або виведення результатів.

Кожен із зазначених етапів є логічно завершеним і може бути реалізований у вигляді окремої функції або програмного компонента, що відповідає модульному принципу побудови програмного забезпечення.

Алгоритм роботи програмного модуля реалізується у вигляді такої детальної послідовності кроків.

1. Початок роботи алгоритму. На початковому етапі виконується ініціалізація програмного модуля, під час якої завантажуються конфігураційні параметри, ініціалізуються необхідні програмні бібліотеки та підготовлюються ресурси для роботи гібридної моделі глибокого навчання. Цей етап забезпечує готовність системи до обробки вхідних даних.

2. Отримання вхідних даних. На вхід алгоритму надходять виконувані файли або їх числові представлення, що підлягають аналізу. Дані можуть передаватися як у вигляді окремих файлів, так і пакетами, що дозволяє реалізувати як одиничний, так і масовий аналіз програмного забезпечення.

3. Перевірка коректності вхідних даних. На даному етапі виконується перевірка формату файлів, їх доступності для читання та відповідності встановленим вимогам. Зокрема перевіряється наявність файлів, їх розмір, коректність структури та відсутність критичних помилок. У разі виявлення некоректних даних формується повідомлення про помилку, після чого алгоритм завершує роботу для поточного зразка, не переходячи до наступних етапів. Такий підхід підвищує надійність програмного модуля та запобігає аварійним завершенням роботи.

4. Попередня обробка даних. Другий етап алгоритму передбачає попередню обробку вхідних даних з метою приведення їх до формату, придатного для аналізу за допомогою гібридної моделі глибокого навчання. На цьому етапі здійснюється зчитування байтового вмісту виконуваних файлів, усунення надлишкової або службової інформації, а також нормалізація даних. Залежно від обраного підходу виконується перетворення байтових послідовностей у двовимірні матриці або зображення фіксованого розміру. Такий спосіб подання даних дозволяє застосовувати згорткові нейронні мережі та зменшує вплив різниці у розмірах виконуваних файлів на результати класифікації.

5. Подання даних на вхід гібридної моделі глибокого навчання. Після завершення передобробки підготовлені дані передаються на вхід гібридної моделі глибокого навчання. Кожна нейронна мережа, що входить до складу гібридної моделі, отримує однакові вхідні дані та виконує їх незалежний аналіз.

6. Формування ознакових представлень. На даному етапі кожна нейронна архітектура формує власний ознаковий вектор, який відображає характерні структурні та статистичні

властивості аналізованого зразка програмного забезпечення. Ознакові представлення формуються на основі глибоких рівнів нейронних мереж і містять узагальнену інформацію про вхідні дані.

7. Об'єднання ознак. Отримані ознакові вектори об'єднуються у спільне узагальнене представлення з використанням механізму агрегації ознак. Такий підхід дозволяє поєднати сильні сторони окремих нейронних архітектур та підвищити інформативність кінцевого ознакового простору.

8. Класифікація зразка. На основі об'єднаного ознакового представлення виконується класифікація аналізованого зразка програмного забезпечення. Алгоритм визначає його належність до класу шкідливого або легітимного програмного забезпечення.

9. Оцінювання рівня впевненості. Паралельно з визначенням класу обчислюється числове значення рівня впевненості прийнятого рішення. Це значення дозволяє оцінити надійність результатів класифікації та може використовуватися для прийняття додаткових рішень у системах захисту інформації.

10. Формування результатів. Результати класифікації подаються у вигляді структурованих даних, що містять інформацію про клас програмного забезпечення та відповідний рівень впевненості моделі.

11. Збереження та/або виведення результатів. Отримані результати зберігаються у файлі або передаються до зовнішніх компонентів системи для подальшого використання, зокрема статистичного аналізу або інтеграції з іншими модулями безпеки.

12. Завершення роботи алгоритму.

Таким чином, розроблений алгоритм роботи програмного модуля забезпечує послідовну, логічно завершену та стійку до помилок обробку даних – від моменту надходження вхідних зразків до формування результатів класифікації. Запропонований алгоритм створює надійну основу для практичної реалізації програмного модуля та подальшого експериментального дослідження його ефективності.

Висновки. У результаті проведеного дослідження розроблено архітектуру програмного модуля виявлення шкідливого програмного забезпечення, яка включає етапи завантаження даних, попередньої обробки, аналізу за допомогою гібридної моделі глибокого навчання, класифікації та формування результатів. Визначено функціональне призначення кожного компонента та забезпечено їх узгоджену взаємодію в межах єдиної системи .

Сформовано алгоритм роботи програмного модуля у вигляді послідовної обробки даних, що охоплює всі етапи – від приймання вхідних зразків до отримання результатів класифікації. Запропонований алгоритм забезпечує логічну цілісність, прозорість функціонування та можливість практичної реалізації програмного модуля.

Встановлено, що використання гібридної моделі глибокого навчання дозволяє ефективно поєднувати ознакові представлення різних нейронних архітектур, що підвищує точність виявлення шкідливого програмного забезпечення та стійкість до модифікацій програмного коду.

Список літератури

1. Nataraj L., Karthikeyan S., Jacob G., Manjunath B. Malware images: Visualization and automatic classification. Proceedings of the International Symposium on Visualization for Cyber Security. 2011. P. 1–7.

2. Saxe J., Berlin K. Deep neural network based malware detection using two dimensional binary program features. Proceedings of the International Conference on Malicious and Unwanted Software. 2015. P. 11–20.

3. Raff E., Barker J., Sylvester J., Brandon R., Catanzaro B., Nicholas C. Malware detection by eating a whole executable. Proceedings of the AAAI Conference on Artificial Intelligence. 2018. Vol. 32, no. 1. P. 268–276.

4. Ucci D., Aniello L., Baldoni R. Survey of machine learning techniques for malware analysis. Computers & Security. 2019. Vol. 81. P. 123–147.

Грицюк Г.І.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

МЕТОДИ МАШИННОГО НАВЧАННЯ ТА АНАЛІЗУ ВЕЛИКИХ ДАНИХ ДЛЯ КЛАСИФІКАЦІЇ ПРОГРАМНИХ ПРОЄКТІВ

Вступ. Стрімка цифровізація економіки, управління та виробничих процесів зумовила суттєве зростання кількості програмних проєктів, їх складності, різноманітності технологічних стеків і обсягів супровідних даних. У сучасній практиці програмної інженерії програмний проєкт уже не розглядається лише як сукупність вихідного коду та технічної документації. Він являє собою багатовимірний об'єкт, що характеризується набором технічних, організаційних і процесних ознак: архітектурними рішеннями, мовами програмування, моделями взаємодії компонентів, способами керування розробленням, інтенсивністю змін, структурою репозиторію, метриками якості та іншими параметрами [1, 2]. За таких умов зростає потреба в інструментах, здатних автоматизовано аналізувати програмні проєкти та відносити їх до певних класів на підставі сукупності релевантних характеристик.

Задача класифікації програмних проєктів є важливою з кількох причин. По-перше, її розв'язання дає змогу впорядковувати великі масиви проєктних даних, що особливо актуально для організацій, які одночасно підтримують значну кількість програмних продуктів. По-друге, результати класифікації можуть бути використані для подальшого ухвалення рішень щодо вибору підходів до розроблення, оцінювання складності, прогнозування ризиків, формування команд, планування тестування та супроводу. По-третє, автоматизована класифікація створює основу для побудови інтелектуальних систем підтримки прийняття рішень у сфері програмної інженерії [1, 2].

Постановка задачі. Проведений аналіз предметної області та існуючих підходів засвідчив, що сучасна програмна інженерія характеризується значним обсягом даних, які супроводжують програмні проєкти на всіх етапах їх життєвого циклу. Водночас наявність великої кількості технічних, структурних і процесних характеристик ускладнює їх ручне впорядкування та об'єктивне віднесення до певних класів [1, 2]. За таких умов актуалізується задача створення програмного засобу, здатного автоматизувати класифікацію програмних проєктів на основі методів машинного навчання та аналізу великих даних [3-5].

Об'єктом дослідження є процес автоматизованої класифікації програмних проєктів.

Предметом дослідження є методи машинного навчання, засоби аналізу великих даних та програмні механізми реалізації модуля класифікації програмних проєктів.

Метою роботи є дослідження методів машинного навчання та аналізу великих даних для класифікації програмних проєктів, що забезпечить автоматизоване віднесення проєкту до заданого класу за сукупністю визначених ознак.

Основний матеріал. З огляду на поставлену задачу найбільш доцільною є багатокомпонентна архітектура, у якій можна виділити п'ять основних рівнів: рівень введення даних, рівень попередньої обробки, рівень машинного навчання, рівень керування результатами та рівень взаємодії з користувачем. Такий підхід дає змогу логічно розмежувати функції системи та спростити її реалізацію. Крім того, це забезпечує можливість окремого тестування кожного компонента та подальшого удосконалення без повної перебудови модуля.

На рівні введення даних передбачається отримання відомостей про програмний проєкт. Джерелом таких даних може бути файл із підготовленою вибіркою, таблиця з ознаками, форма ручного введення або інший структурований формат подання інформації. У межах бакалаврської роботи найбільш практичним є використання табличного представлення даних, наприклад у форматі CSV, оскільки воно є простим у реалізації, добре підтримується засобами Python і безпосередньо узгоджується з бібліотеками аналізу даних [3-5]. Основна функція цього компонента полягає в завантаженні, первинній перевірці та передаванні даних до наступного етапу обробки.

Рівень попередньої обробки виконує критично важливу роль, оскільки саме на ньому сирі

вхідні дані перетворюються у форму, придатну для навчання моделі. До функцій цього рівня належать очищення даних, усунення або заповнення пропусків, кодування категоріальних ознак, нормалізація або масштабування числових значень, а також формування підсумкової матриці ознак. Необхідність такого етапу зумовлена тим, що характеристики програмних проєктів мають різну природу та не можуть бути безпосередньо подані в модель без попереднього узгодження форматів [3]. Крім того, саме на цьому рівні може виконуватися поділ вибірки на навчальну та тестову частини.

Рівень машинного навчання є центральним у структурі програмного модуля. Він відповідає за створення, навчання та використання моделі класифікації. У межах даної роботи як основний алгоритм обрано випадковий ліс, а тому відповідний компонент повинен реалізовувати ініціалізацію моделі, налаштування її параметрів, навчання на підготовленій вибірці та отримання прогнозів для нових об'єктів [3]. Якщо в системі передбачається порівняння кількох моделей, цей самий рівень може містити підмодулі для базових алгоритмів, наприклад логістичної регресії або дерева рішень. Такий підхід дозволяє реалізувати архітектуру, у якій модель машинного навчання є змінним компонентом, а не жорстко вбудованим елементом усієї системи.

Рівень керування результатами призначений для оброблення та збереження результатів роботи моделі. Він забезпечує виведення передбаченого класу програмного проєкту, обчислення основних метрик якості під час тестування, формування зведеної інформації про точність класифікації та, за потреби, збереження навченої моделі для подальшого повторного використання. Практичне значення цього рівня полягає в тому, що він поєднує математичний результат моделювання з прикладним сценарієм використання програмного модуля. Без цього компонента система залишалася б лише набором процедур навчання, а не завершеним прикладним рішенням [3].

Рівень взаємодії з користувачем забезпечує подання вхідних даних і відображення результатів роботи програми. Він може бути реалізований у вигляді консольного інтерфейсу, форми введення або простого графічного інтерфейсу. Його головне призначення полягає в тому, щоб надати користувачеві можливість завантажити або ввести дані про проєкт, ініціювати процедуру класифікації та отримати підсумковий результат у зрозумілому вигляді. При цьому архітектурно важливо, щоб інтерфейсний рівень не містив усієї бізнес-логіки, а лише взаємодіяв із відповідними внутрішніми компонентами системи [6, 7].

Загальна логіка архітектури програмного модуля може бути подана як послідовний ланцюг перетворення даних: введення даних → підготовка даних → навчання або завантаження моделі → класифікація → подання результату.

У межах проєктування архітектури доцільно також визначити основні потоки даних між компонентами. Спочатку дані про програмні проєкти надходять у систему через модуль введення. Після цього вони передаються до модуля попередньої обробки, де формуються уніфіковані ознакові вектори. Далі ці дані можуть бути використані у двох режимах: у режимі навчання вони надходять до модуля навчання моделі, а в режимі практичної класифікації - до модуля класифікації, який використовує вже навчений класифікатор. У будь-якому з цих випадків результати передаються до модуля оцінювання та модуля подання результатів. Такий розподіл потоків дозволяє відокремити етап створення моделі від етапу її прикладного використання.

На рисунку 1 відображено базову архітектурну логіку роботи системи. Така схема є достатньо простою для реалізації, але водночас охоплює всі основні процеси, необхідні для функціонування програмного модуля.

З точки зору програмної реалізації доцільно орієнтуватися на модульний підхід, у якому кожен архітектурний блок відповідає окремому програмному файлу, класу або групі функцій. Наприклад, можуть бути виділені такі логічні модулі: data_loader, preprocessing, model_training, prediction, evaluation, ui. Такий підхід відповідає принципам розподілу відповідальності в програмній інженерії та спрощує подальше супроводження програмного коду [6, 7]. Крім того, він дозволяє тестувати та налагоджувати окремі частини системи незалежно одна від одної.

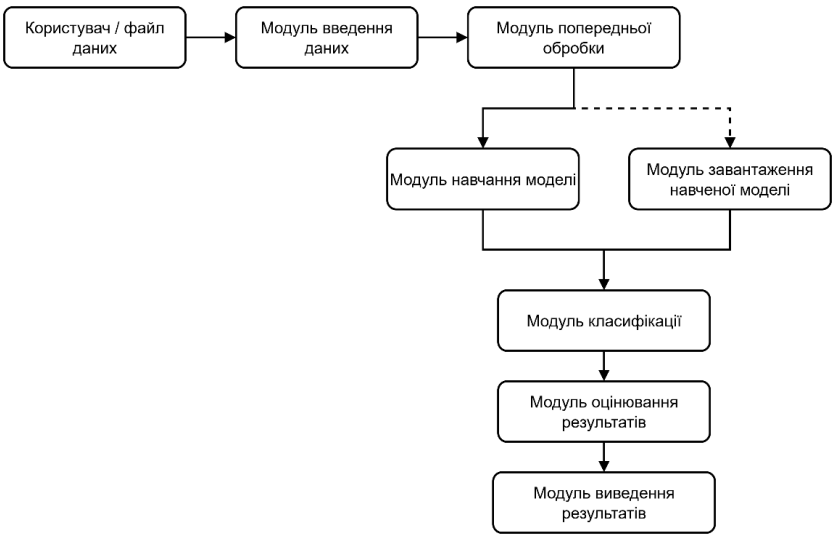


Рисунок 1 – Архітектура програмного модуля класифікації програмних проєктів

У контексті архітектурного проєктування доцільно також визначити основні сценарії функціонування системи. Перший сценарій пов'язаний із навчанням моделі. У цьому випадку користувач або розробник завантажує набір даних, виконує їх попередню обробку, запускає навчання моделі та отримує результати її тестування. Другий сценарій пов'язаний із класифікацією нового проєкту. У цьому режимі в систему подаються характеристики окремого програмного проєкту, після чого виконується їх перетворення за тими самими правилами, що були використані під час навчання, і система повертає передбачений клас. Таке розмежування сценаріїв дозволяє побудувати архітектуру, у якій дослідницький та прикладний режими використання не змішуються, але працюють на спільній основі.

З позицій надійності системи важливо, щоб архітектура враховувала можливість перевірки коректності вхідних даних. У реальних умовах помилки можуть виникати через пропуски значень, невідповідність формату, некоректні типи ознак або неповне введення характеристик нового проєкту. Тому доцільно передбачити в модулі введення та попередньої обробки процедури валідації, які запобігатимуть передаванню некоректних даних до моделі машинного навчання. Хоча це ускладнює реалізацію, така вимога є виправданою, оскільки підвищує практичну придатність програмного засобу.

Окремої уваги потребує питання збереження навченої моделі. Якщо щоразу виконувати повне перенавчання перед класифікацією нового проєкту, це зменшить ефективність роботи системи. Тому архітектура повинна передбачати механізм збереження моделі після завершення навчання та її повторного завантаження під час експлуатації. Для цього можуть бути використані стандартні засоби серіалізації, що підтримуються мовою Python та бібліотеками машинного навчання. Завдяки цьому модуль може працювати у двох незалежних режимах: режимі створення моделі та режимі її застосування.

З погляду загальної логіки побудови програмного забезпечення обрана архітектура може бути охарактеризована як функціонально-модульна з елементами шарового підходу. Її верхній рівень представлений засобами взаємодії з користувачем, середній - логікою підготовки даних і використання моделі, а нижній - інструментами роботи з даними та збереження результатів. Такий підхід є достатньо простим для реалізації в навчальному проєкті й водночас забезпечує структурованість рішення, що відповідає принципам проєктування якісного програмного забезпечення [6, 7].

Висновки. Обґрунтовано актуальність використання методів машинного навчання та аналізу великих даних для класифікації програмних проєктів. Встановлено, що сучасний програмний проєкт є складним багатовимірним об'єктом, який характеризується не лише вихідним кодом, а й архітектурними рішеннями, технологічним стеком, процесами розроблення, структурою репозиторію, інтенсивністю змін та іншими технічними й організаційними параметрами . Саме тому автоматизована класифікація таких проєктів є

доцільною для впорядкування проєктних даних, підтримки прийняття рішень, оцінювання складності, прогнозування ризиків і вибору підходів до розроблення.

У роботі визначено, що основою для побудови відповідного програмного засобу має бути багатокомпонентна архітектура. Вона включає рівень введення даних, рівень попередньої обробки, рівень машинного навчання, рівень керування результатами та рівень взаємодії з користувачем. Така структура дозволяє логічно розмежувати функції системи, спростити її реалізацію, забезпечити окреме тестування компонентів і створити передумови для подальшого розширення функціональності модуля.

Особливу увагу приділено етапу попередньої обробки даних, оскільки характеристики програмних проєктів мають різну природу та потребують приведення до єдиного формату. У матеріалі показано, що перед навчанням моделі необхідно виконати очищення даних, оброблення пропусків, кодування категоріальних ознак, нормалізацію числових значень і формування матриці ознак . Це є важливою умовою коректної роботи класифікаційної моделі та забезпечення достовірності отриманих результатів.

У межах запропонованого підходу центральним компонентом системи визначено рівень машинного навчання. Основним алгоритмом обрано випадковий ліс, оскільки він добре працює з табличними даними, є стійким до неоднорідності ознак і може забезпечувати якісну класифікацію програмних проєктів. Також передбачено можливість порівняння з базовими моделями, зокрема логістичною регресією або деревом рішень, що підвищує обґрунтованість вибору основного методу .

Запропонована архітектура підтримує два основні сценарії функціонування: навчання моделі та класифікацію нового програмного проєкту. У першому сценарії система завантажує набір даних, виконує їх підготовку, навчає модель і оцінює її якість. У другому сценарії користувач подає характеристики нового проєкту, після чого система виконує їх перетворення та повертає передбачений клас . Таке розмежування режимів роботи підвищує практичну придатність програмного модуля та дозволяє використовувати його як у дослідницькому, так і в прикладному режимі.

Важливим результатом є обґрунтування необхідності збереження навченої моделі та повторного її використання без постійного перенавчання. Це підвищує ефективність роботи системи, скорочує час класифікації нових проєктів і забезпечує стабільність результатів. Крім того, передбачення процедур валідації вхідних даних підвищує надійність програмного засобу та запобігає передаванню некоректної інформації до моделі машинного навчання

Список літератури

1. Gurcan F., Cagiltay N. E. Big Data Software Engineering: Analysis of Knowledge Domains and Skill Sets Using LDA-Based Topic Modeling // *IEEE Access*. 2019. Vol. 7. P. 82541–82552.
2. Wijesiriwardana C., Wimalaratne P. Software Engineering Data Analytics: A Framework Based on a Multi-Layered Abstraction Mechanism // *IEICE Transactions on Information and Systems*. 2019. Vol. E102.D, no. 3. P. 637–639.
3. James G., Witten D., Hastie T., Tibshirani R., Taylor J. *An Introduction to Statistical Learning: With Applications in Python*. Cham : Springer, 2023. DOI: 10.1007/978-3-031-38747-0.
4. Гавриленко О. В., Онищенко В. В., Мягкий М. Ю. Методи аналізу даних та машинного навчання в інформаційно-управляючих системах. Київ : КПІ ім. Ігоря Сікорського, 2025. 318 с.
5. Гороховатський В.О., Творошенко І.С. Методи інтелектуального аналізу та оброблення даних: навч. посібник. Харків: ХНУРЕ, 2021. 92 с.
6. Ramírez-Noriega A., Martínez-Ramírez Y., Figueroa-Pérez J. F., et al. inDev: A Software to Generate an MVC Architecture Based on the ER Model // *Computer Applications in Engineering Education*. 2022. Vol. 30, no. 1. P. 259–274.
7. Singh I., Lee S. W. RE_BBC: Requirements Engineering in a Blockchain-Based Cloud: Its Role in Service-Level Agreement Specification // *IEEE Software*. 2020. Vol. 37, no. 5. P. 7–12.

Заєць О.Ю.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

ПРОГРАМНА СИСТЕМА АНАЛІЗУ МЕРЕЖЕВИХ ЛОГІВ З ВИКОРИСТАННЯМ NLP-ПІДХОДУ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ

Вступ. Стрімке зростання обсягів мережевого трафіку, ускладнення архітектури інформаційно-комунікаційних систем і постійна еволюція кіберзагроз зумовили потребу в розробленні ефективних засобів автоматизованого аналізу мережевих подій. У сучасних умовах мережеві логи є одним із головних джерел інформації про стан інфраструктури, характер взаємодії між вузлами мережі, спроби несанкціонованого доступу, відхилення від типової поведінки та ознаки потенційних атак. Саме тому своєчасне виявлення аномалій у логах набуло важливого значення для забезпечення кібербезпеки, підтримання працездатності сервісів і зниження ризику порушення цілісності, конфіденційності та доступності даних [1, 2].

Постановка задачі. Проведений аналіз предметної області та існуючих підходів показав, що мережеві логи є цінним, але складним джерелом даних для виявлення аномалій. Основна проблема полягає в тому, що логові записи надходять із різних джерел, мають напівструктурований формат, містять службові поля, текстові фрагменти, числові параметри та часові мітки. За таких умов традиційні сигнатурні та статистичні методи не завжди забезпечують належну гнучкість, а класичні ML-підходи потребують якісного формування ознак. Це зумовлює доцільність побудови програмної системи, у якій мережеві логи розглядаються як спеціалізовані текстові об’єкти, придатні для подальшого NLP-аналізу.

У даної роботи досліджується задача автоматизованого виявлення аномалій у мережевих логах на основі їх попереднього оброблення, текстового представлення та подальшої класифікації за допомогою NLP-підходу. Практичний зміст цієї задачі полягає у створенні програмної системи, здатної приймати набір логових записів, перетворювати їх у стандартизоване подання, аналізувати ознаки подій і формувати результат у вигляді класифікації запису як нормального або аномального.

Таким чином, основну задачу дослідження сформовано так: розробити програмну систему аналізу мережевих логів, яка забезпечує автоматизоване виявлення аномальних подій на основі NLP-підходу та дає змогу оцінювати якість прийнятих рішень за стандартними метриками класифікації.

Об’єктом дослідження є процес аналізу мережевих логів у задачах виявлення аномалій у комп’ютерних мережах.

Предметом дослідження є методи, моделі та програмні засоби оброблення мережевих логів на основі NLP-підходу для автоматизованого виявлення аномальних записів.

Метою – є розроблення програмної системи аналізу мережевих логів з використанням NLP-підходу для виявлення аномальних подій у мережевому трафіку.

Основний матеріал. Архітектуру програмної системи сформовано з урахуванням послідовності оброблення мережевих логів: від надходження вхідних записів до отримання результату класифікації. Основна вимога до такої архітектури полягає у відокремленні етапів підготовки даних, текстового подання, NLP-аналізу та формування результатів. Це спрощує реалізацію, тестування та подальше вдосконалення окремих модулів системи.

Програмну систему доцільно побудувати як модульну структуру, що складається з п’яти основних компонентів:

- модуля завантаження логів;
- модуля попереднього оброблення;
- модуля текстового подання логів;
- модуля NLP-класифікації;
- модуля формування та відображення результатів.

Модуль завантаження логів забезпечує приймання вхідних даних із файлу або підготовленого набору записів. На цьому етапі логові дані зчитуються, перетворюються у

внутрішню структуру зберігання та передаються до наступного модуля. Основне завдання цього компонента полягає в тому, щоб уніфікувати початковий формат записів і підготувати їх до подальшого оброблення.

Модуль попереднього оброблення виконує очищення та нормалізацію даних. На цьому етапі усуваються порожні або некоректні значення, узгоджуються назви полів, видаляються дублікати, за потреби виконуються перетворення часових міток, адрес, портів та інших атрибутів. Саме цей модуль формує впорядкований набір записів, придатний для подальшого аналізу.

Модуль текстового подання є ключовим елементом архітектури. Його призначення полягає у перетворенні структурованого або напівструктурованого логового запису в текстовий шаблон. У результаті кожна мережева подія подається як цілісний текстовий опис, що містить основні атрибути запису в уніфікованому форматі. Такий підхід дає змогу застосувати методи NLP-аналізу до логів як до спеціалізованих текстових об’єктів.

Модуль NLP-класифікації виконує основне аналітичне завдання системи. На його вхід надходять текстові представлення логів, які після токенізації або векторизації подаються в модель класифікації. Результатом роботи модуля є визначення класу події: нормальна або аномальна. У межах архітектури цей компонент є центральним, оскільки саме він реалізує інтелектуальне виявлення відхилень у логових даних.

Модуль формування та відображення результатів забезпечує подання отриманих висновків у зручному вигляді. У ньому зберігаються результати класифікації, обчислюються підсумкові показники якості та формуються повідомлення для користувача. Для експериментального сценарію достатньо табличного відображення записів, передбаченого класу та узагальнених метрик.

Загальну архітектуру системи представлено на рисунку 1.

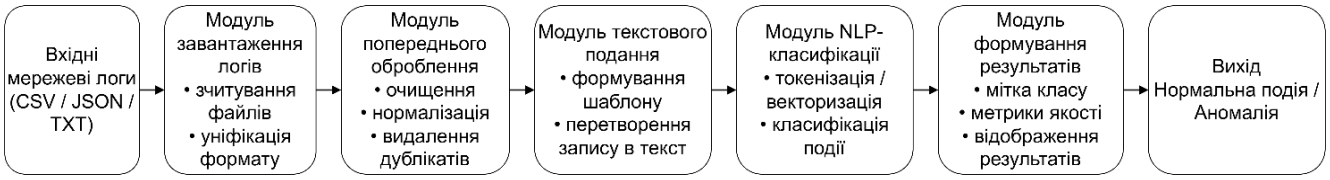


Рисунок 1 – Архітектура програмної системи аналізу мережевих логів

Взаємодія модулів є послідовною. Кожен наступний компонент отримує результат роботи попереднього. Така схема є простою, зрозумілою та придатною для прототипної реалізації. Крім того, вона дає змогу окремо вдосконалювати кожен модуль без зміни всієї системи. Наприклад, можна змінити спосіб текстового представлення логів або модель класифікації без перебудови інших частин архітектури.

Отже, архітектуру програмної системи сформовано як модульну послідовну структуру, орієнтовану на оброблення мережевих логів, їх перетворення у текстове подання та подальшу NLP-класифікацію.

За результатами тестування система продемонструвала високі показники якості бінарної класифікації (таблиця 1).

Таблиця 1 – Основні результати експериментального оцінювання

Метрика	Значення, %
Accuracy	97,5
Precision	96,4
Recall	97,8
F1-score	97,1
FPR	2,7
FNR	2,3

Отримані результати свідчать, що система забезпечує високу частку правильних рішень і добре виявляє аномальні події. Найбільш показовим є значення recall 97,8 %, оскільки воно характеризує здатність системи знаходити аномалії та не пропускати небезпечні записи. Водночас precision 96,4 % підтверджує, що кількість хибних спрацьовувань залишається невисокою.

Для наочного подання результатів сформувати матрицю помилок (рисунок 2).

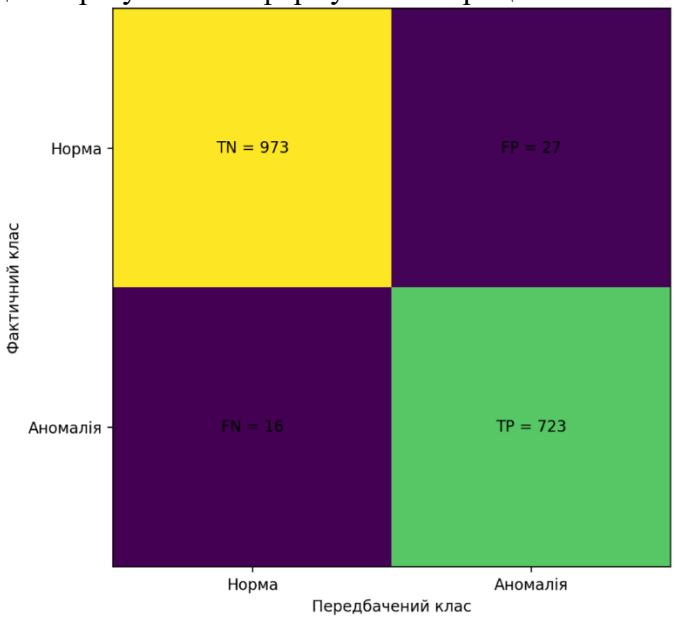


Рисунок 2 – Матриця помилок результатів класифікації

Подана матриця відповідає загальним отриманим показникам і дає змогу краще оцінити структуру помилок системи. Із рисунку 2 видно, що основна частина записів класифікується правильно. Кількість помилково пропущених аномалій є меншою, ніж кількість хибних тривог, що є позитивною характеристикою для системи виявлення вторгнень.

Окремо доцільно порівняти отриманий результат з кількома поширеними підходами, згаданими в огляді: Decision Tree, CNN-BiLSTM та GBM [3-5]. Узагальнене порівняння наведено в таблиці 2.

Таблиця 2 – Порівняння результатів із базовими підходами

Метод	Accuracy, %	Precision, %	Recall, %	F1-score, %
Decision Tree	91,2	90,4	89,8	90,1
GBM	94,8	94,0	93,5	93,7
CNN-BiLSTM	95,9	95,1	95,4	95,2
Запропонована система	97,5	96,4	97,8	97,1

Як видно з таблиці 2, запропонована система демонструє найкращі підсумкові показники серед розглянутих підходів. Основна перевага полягає не лише в підвищенні accuracy, а й у зростанні recall, тобто у кращому виявленні аномальних подій. Для задач кібербезпеки це є важливішим за незначне скорочення окремих допоміжних показників.

Висновки. У межах дослідження сформульовано задачу розроблення програмної системи, яка забезпечує завантаження, попереднє оброблення, текстове подання та NLP-класифікацію мережевих логів. Запропоновано модульну архітектуру системи, до складу якої включено модулі завантаження логів, попереднього оброблення, текстового подання, NLP-класифікації та формування результатів. Така структура забезпечила послідовну та зрозумілу організацію процесу аналізу мережевих подій.

Експериментальні результати підтвердили ефективність запропонованого підходу. Встановлено, що система забезпечує високі показники якості бінарної класифікації: accuracy 97,5 %, precision 96,4 %, recall 97,8 % та F1-score 97,1 %. Отримані значення свідчать про високу

частку правильних рішень, незначну кількість хибних спрацьовувань і добру здатність системи виявляти аномальні події. Аналіз матриці помилок також показав, що основна частина записів класифікується правильно, а кількість пропущених аномалій залишається низькою.

Порівняння із базовими підходами, зокрема Decision Tree, GBM і CNN-BiLSTM, показало, що запропонована система має кращі підсумкові результати. Найважливішою перевагою є підвищення recall, тобто покращення здатності виявляти аномальні мережеві події, що має особливе значення для задач кібербезпеки. Це підтвердило доцільність використання NLP-підходу, за якого мережевий лог розглядається як цілісний текстовий опис події.

Отже, поставлену задачу розв'язано: розроблено концепцію програмної системи аналізу мережевих логів, обґрунтовано її архітектуру та підтверджено ефективність запропонованого підходу експериментальними результатами. Отримані висновки засвідчили, що використання NLP-підходу є перспективним напрямом для побудови інтелектуальних систем виявлення аномалій у мережевих логах.

Список літератури

1. Hubballi N., Suryanarayanan V. False alarm minimization techniques in signature-based intrusion detection systems: a survey. *Computer Communications*. 2014. Vol. 49. P. 1–17.

2. Masdari M., Khezri H. A survey and taxonomy of the fuzzy signature-based intrusion detection systems. *Applied Soft Computing*. 2020. Vol. 92. Article 106301.

3. Louk M. H. L., Tama B. A. Dual-IDS: a bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Systems with Applications*. 2023. Vol. 213. Article 119030.

4. Rai K., Syamala Devi M., Guleria A. Decision tree based algorithm for intrusion detection. *International Journal of Advanced Networking and Applications*. 2016. Vol. 7, no. 4. P. 2828–2834.

5. Sinha J., Manollas M. Efficient deep CNN-BiLSTM model for network intrusion detection. In: *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*. 2020.

Шорін В.С.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

АНСАМБЛЕВІ МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ У СМАРТ-СИСТЕМІ ОБЛІКУ

Вступ. Смарт-системи обліку застосовуються для автоматизованого збору показників споживання електроенергії, газу чи води та подальшого використання цих даних у розрахунках і керуванні інфраструктурою [1]. З технічного погляду такі рішення поєднують лічильники, канали зв'язку, серверні компоненти та програмні підсистеми, що забезпечують приймання, збереження й аналіз вимірювань [2]. Практика впровадження подібних систем демонструє, що ключовим ресурсом стають саме дані: вони надходять потоково, накопичуються у великих обсягах, потребують очищення та контролю якості [3, 4].

Постановка задачі. Результати аналізу предметної області та класифікації аномалій і загроз свідчать, що дані смарт-обліку є потоковими, часовими та схильними до помилок як технічного, так і безпекового характеру. За таких умов завдання виявлення аномалій доцільно формулювати не як одноразову "перевірку файлу", а як програмний модуль, який може бути вбудований у контур збору та аналізу вимірювань і працювати з регулярними надходженнями показників. Оскільки виявлення аномалій часто використовується як елемент моніторингу та реагування, важливо заздалегідь визначити, що саме вважати "аномалією", які дані є на вході, який формат результату потрібен на виході, а також якими метриками оцінюватиметься ефективність рішення [1, 4].

Задача дослідження полягає в тому, щоб проєктувати прототип програмного модуля, який на основі даних смарт-лічильників автоматично визначає підозрілі або нетипові

вимірювання/інтервали та формує рішення про наявність аномалії. Враховуючи поширені сценарії викривлення даних, модуль має бути орієнтований як на виявлення збоїв якості даних (пропуски, дублікати, неможливі значення), так і на виявлення поведінкових відхилень, що можуть бути пов'язані з маніпуляціями або крадіжками ресурсу [5].

Об'єктом дослідження є процеси збору та аналізу даних у смарт-системах обліку. Предметом дослідження є методи та програмні засоби виявлення аномалій у даних смарт-обліку з використанням ансамблевих моделей машинного навчання. Метою дослідження є проєктування програмного модуля виявлення аномалій у смарт-системі обліку на основі ансамблевих методів машинного навчання та оцінювання його ефективності на даних обліку.

Основний матеріал. Побудова програмного модуля виявлення аномалій у смарт-системі обліку доцільна як реалізація послідовного конвеєра обробки даних: від надходження показників лічильників до формування рішення та подальшого використання результату в моніторингу [4]. Такий підхід відповідає природі смарт-обліку, де дані надходять регулярно, мають часовий контекст і можуть містити пропуски або викривлення [1]. Додатково враховується вимога до масштабованості, оскільки обсяги вимірювань у великих системах набувають ознак "big data".

Запропонований підхід ґрунтується на п'яти послідовних етапах:

1. Приймання даних.
2. Первинний контроль якості та очищення.
3. Формування ознак.
4. Детектування аномалій.
5. Фіксація результату та інтеграція з моніторингом.

Кожен етап має власну роль у зменшенні помилок і підвищенні стійкості детектора. Зокрема, від якості формування ознак залежить здатність моделі відрізняти природні коливання споживання від підозрілих відхилень, тоді як очищення даних зменшує кількість "хибних" аномалій, що виникають через технічні дефекти потоку [4].

На рисунку 1 представлено узагальнену архітектуру підходу у вигляді потоку даних між логічними компонентами.

Рисунок 1 – Архітектура підходу до виявлення аномалій у смарт-обліку

Компоненти та їх функції:

1) Джерело даних (смарт-лічильники та допоміжні компоненти). На вході системи знаходяться смарт-лічильники та інфраструктура передавання показників. Дані містять ідентифікатор точки обліку, часову мітку, значення споживання, а за наявності – службові ознаки стану чи якості каналу [1, 4]. У рамках роботи припускається, що ці дані надходять у вигляді потоку записів або пакета записів за певний період.

2) Компонент приймання. Залежно від сценарію реалізації приймання може здійснюватися:

– пакетно (читання файлів/таблиць);
– потоково через брокер повідомлень (наприклад, Kafka) або інші механізми доставки подій.

Потоковий підхід є типовим для сервісної архітектури великих даних, де дані надходять у реальному часі та мають оброблятися без накопичення великих затримок [4]. У разі використання Spark Structured Streaming можливе формування мікропакетів та обробка за принципом "near real-time".

3) Первинний контроль якості та очищення. На цьому етапі виконуються прості й швидкі перевірки: коректність типів даних, часова впорядкованість, видалення дублікатів, заповнення або маркування пропусків, фільтрація очевидно некоректних значень. Мета етапу полягає не в “пошуку аномалій” у змістовому сенсі, а в усуненні технічного сміття, яке інакше спричинить зайві спрацювання детектора. Така логіка узгоджується з практикою використання правил та фільтрацій як базового шару контролю [4, 5].

4) Формування ознак. Оскільки сирі вимірювання є часовими рядами, їх доцільно перетворювати на набір ознак, які відображають контекст. Зазвичай використовуються агрегати за часовими вікнами (середнє, медіана, стандартне відхилення), прирости (різниця між поточним і попереднім значенням), показники стабільності, частка нульових значень, індикатори “надмірної регулярності” тощо [4]. Такий підхід дозволяє подати задачу у формі табличного набору ознак, для якого деревові ансамблі є природним вибором.

5) Модель детектування (ансамблевий класифікатор/детектор). На цьому етапі застосовується навчена модель, наприклад Random Forest [6] або бустингові дерева (Gradient Boosting, XGBoost, LightGBM) [7-9]. Виходом є скор (оцінка ризику) і/або мітка “аномалія/норма”. У прикладних задачах, зокрема для виявлення енергетичних крадіжок, ансамблеві моделі розглядаються як практичний інструмент через стійкість до шуму та здатність уловлювати складні комбінації ознак.

6) Компонент результатів: журналювання, сповіщення, інтеграція. Рішення модуля має бути зафіксоване у журналі подій із мінімально необхідним набором полів: meter_id, timestamp, скор/мітка, ключові ознаки, службова інформація [4]. За потреби формується сповіщення для оператора або автоматизованої системи реагування. З позиції кібербезпеки та управління ризиками важливо забезпечити аудит подій і можливість розслідування причин спрацювань.

У рамках роботи можливі два режими, які відрізняються переважно способом приймання даних:

1. Пакетний режим. Дані зчитуються з файлу або бази даних за визначений період; формуються ознаки; виконується детектування; формується звіт або список аномалій. Такий режим простіше реалізувати й перевірити експериментально, зберігаючи коректність постановки задачі.
2. Потоковий режим. Дані надходять безперервно; ознаки обчислюються у ковзних вікнах; рішення формується для кожного запису або мікропакета. Для реалізації цього режиму зазвичай використовуються брокери повідомлень та стримінгові фреймворки. Перевага полягає в оперативності, однак зростають вимоги до синхронізації часу, обліку затримок і обробки пропусків.

Отже, архітектура підходу базується на конвеєрі “дані – якість – ознаки – ансамбль – рішення”, який відповідає природі смарт-обліку та дозволяє поєднати інженерні засоби контролю якості з ML-детектуванням. Така організація підвищує стійкість до шуму та пропусків, а ансамблеві методи забезпечують практичний компроміс між якістю, стабільністю та придатністю до впровадження у реальних умовах обліку.

Висновки. Запропонований підхід базується на послідовному конвеєрі обробки даних: приймання показників, первинний контроль якості, очищення, формування ознак, застосування ансамблевої моделі та фіксація результату. Така структура є логічною, оскільки кожен етап виконує окрему функцію: очищення зменшує кількість хибних спрацювань, формування ознак перетворює часові ряди на зручний для аналізу табличний формат, а модель детектування оцінює підозрілість вимірювання. Особливу увагу приділено формуванню ознак, оскільки саме вони дозволяють відобразити поведінку споживання у зрозумілому для моделі вигляді. Доцільними визначено ознаки, що описують середні значення, медіану, стандартне відхилення, прирости, стабільність, частку нульових значень та інші характеристики часових вікон. Це дає змогу відрізняти природні коливання споживання від нетипових відхилень. Для детектування аномалій обґрунтовано використання ансамблевих методів машинного навчання, зокрема Random Forest, Gradient Boosting, XGBoost та LightGBM. Такі моделі є придатними для роботи з табличними ознаками, мають достатню стійкість до шуму та можуть виявляти складні комбінації факторів, які не завжди помітні за допомогою простих правил або порогових

перевірок. Також визначено два можливі режими роботи модуля: пакетний і потоковий. Пакетний режим є простішим для реалізації та експериментальної перевірки, тоді як потоковий режим краще відповідає умовам реального смарт-обліку, де дані надходять безперервно. Водночас потокова реалізація потребує додаткової уваги до синхронізації часу, затримок, пропусків і стабільності обробки.

Список літератури

1. Kabalci Y. A survey on smart metering and smart grid communication. Renewable and Sustainable Energy Reviews. 2016. Vol. 57. P. 302–318.
2. Zeng Yan, Ye Xiaoying, Wang Kang, Gao Yanhao, Zheng Hongliang. Design of an intelligent unattended metering system. Industrial Technology and Vocational Education. 2024. Vol. 22, no. 5. P. 1–5.
3. Ji X., Li T. Analysis on Safety Performance Improvement of Smart Metering System Based on Big data. Procedia Computer Science. 2025. Vol. 262. P. 909–918.
4. Wang J., Yang Y., Wang T., et al. Big data service architecture: a survey. Journal of Internet Technology. 2020. Vol. 21, no. 2. P. 393–405.
5. Zhang Yongmin, Liu Shengnan, Yi Ling, An Jiyuan. Power metering system based on filtering to avoid power theft behavior dynamic identification method. Mechanical Design and Manufacturing Engineering. 2024. Vol. 53, no. 3. P. 129–132.
6. Breiman L. Random Forests. Machine Learning. 2001. Vol. 45, no. 1. P. 5–32.
7. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 785–794.
8. Ke G., Meng Q., Finley T., et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems (NeurIPS). 2017.
9. Friedman J. H. Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics. 2001. Vol. 29, no. 5. P. 1189–1232.

Ярунів М.А.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ТА ПАРАЛЕЛЬНОЇ ОБРОБКИ ФІНАНСОВИХ ДАНИХ

Вступ. Цифрова трансформація фінансового сектору супроводжується стрімким зростанням обсягів даних, які формуються під час виконання платіжних операцій, бухгалтерського та управлінського обліку, бюджетування, ризик-менеджменту, взаємодії з клієнтами та зовнішніми інформаційними ресурсами.

У таких умовах зростає роль технологій Big Data у фінансах, оскільки саме вони забезпечують можливість отримання своєчасних аналітичних висновків на основі великих масивів гетерогенних даних [1]. Водночас упровадження аналітики великих даних у банківських і фінансових установах потребує узгоджених організаційних та технологічних рішень, зокрема щодо управління даними, регламентів доступу, якості та життєвого циклу наборів даних [2, 3].

Постановка задачі. Цифрова трансформація фінансового сектору супроводжується стрімким зростанням обсягів даних, які формуються під час виконання платіжних операцій, бухгалтерського та управлінського обліку, бюджетування, ризик-менеджменту, взаємодії з клієнтами та зовнішніми інформаційними ресурсами. У таких умовах зростає роль технологій Big Data у фінансах, оскільки саме вони забезпечують можливість отримання своєчасних аналітичних висновків на основі великих масивів гетерогенних даних [1]. Водночас упровадження аналітики великих даних у банківських і фінансових установах потребує

узгоджених організаційних та технологічних рішень, зокрема щодо управління даними, регламентів доступу, якості та життєвого циклу наборів даних [2, 3].

Об’єкт дослідження – процеси зберігання та обробки фінансових даних у хмарному та кластерному середовищі.

Предмет дослідження – методи та програмні засоби розподіленого зберігання і паралельної обробки фінансових даних, а також підходи до забезпечення узгодженості, захисту і якості даних.

Мета роботи полягає у проєктуванні програмного модуля розподіленого зберігання та паралельної обробки фінансових даних, який забезпечує масштабованість, відмовостійкість, підвищену продуктивність аналітичних операцій, а також базові механізми захисту та контролю якості даних.

Основний матеріал. Архітектура програмного модуля розподіленого зберігання та паралельної обробки фінансових даних формується з урахуванням вимог продуктивності, масштабованості, відмовостійкості, узгодженості та безпеки [4]. У фінансових системах одночасно існують два типи навантаження: операційне (швидкі вибірки за ключовими атрибутами) та аналітичне (масові сканування, агрегації, консолідація). Поєднання цих навантажень у межах одного централізованого сховища часто призводить до зростання затримок і ускладнює масштабування [4]. Тому доцільним є модульний підхід, у якому розподілене зберігання і паралельна обробка реалізуються як узгоджені компоненти єдиного контуру даних [5].

Програмний модуль розглядається як інфраструктурний компонент між джерелами фінансових даних (ERP/CRM, транзакційні журнали, звітність, зовнішні показники) та споживачами результатів (BI, звітні підсистеми, прикладні сервіси аналітики). На концептуальному рівні модуль реалізує послідовність етапів: приймання даних → валідація/нормалізація → розподілене зберігання → паралельна обробка → публікація результатів [5]. Така модель забезпечує відокремлення операційного та аналітичного контурів і дозволяє зберігати історичні масиви без погіршення доступу до “гарячих” даних.

Архітектура модуля поділяється на три рівні (рисунок 1).

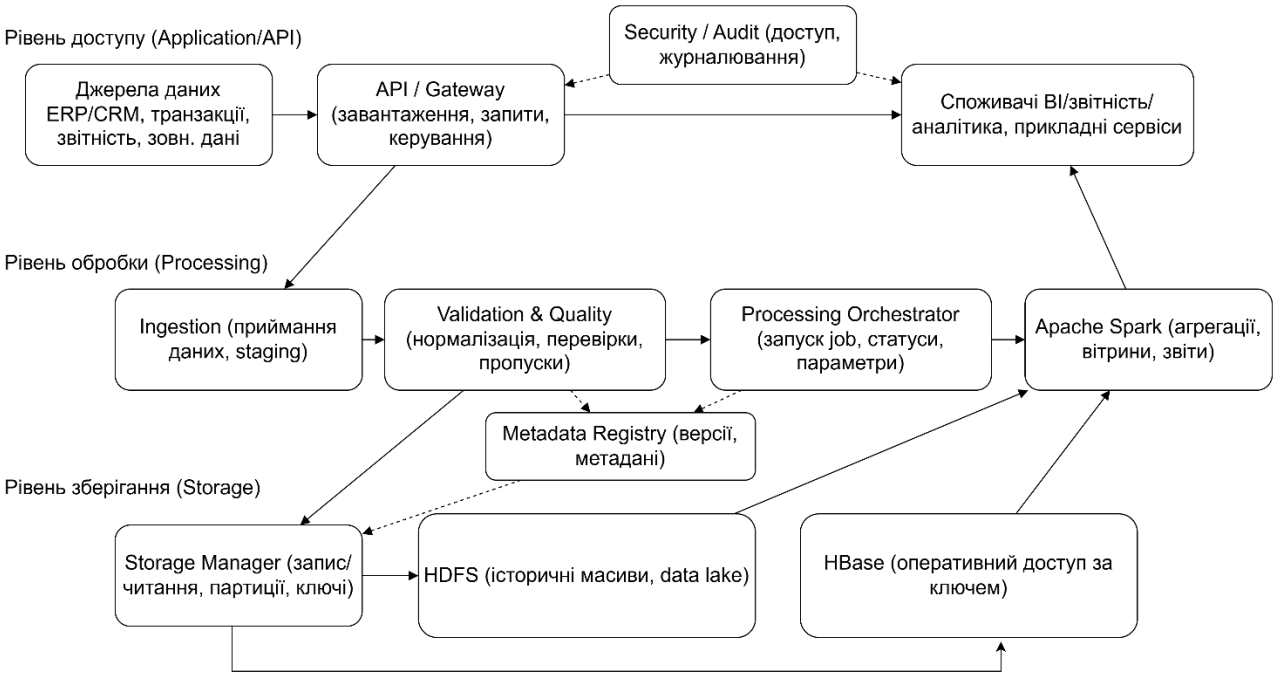


Рисунок 1 – Загальна архітектурна схема програмного модуля: взаємодія рівнів та основних компонентів

1. Рівень зберігання (Storage Layer). Для довготривалого зберігання великих обсягів даних використовується HDFS як розподілена файлова система з блочною організацією та реплікацією. Для оперативного доступу за ключем застосовується HBase як колоночне NoSQL-

сховище поверх HDFS. Така комбінація забезпечує одночасно високу пропускну здатність при скануванні історичних даних і можливість швидких вибірок для сервісних сценаріїв [6, 7].

2. Рівень обробки (Processing Layer). Паралельне виконання обчислювальних задач реалізується на Apache Spark, який підтримує кластерні обчислення, DAG-планування та оптимізацію повторних обробок за рахунок роботи з робочими наборами даних. Spark використовується для агрегацій, формування вітрин, регламентних розрахунків, перевірок якості та підготовки даних [5].

3. Прикладний рівень доступу (Application/API Layer). Призначений для взаємодії із зовнішніми системами: керування завантаженнями, запуск задач, отримання результатів, перегляд статусів і протоколів перевірок. Наявність такого рівня узгоджується з практикою впровадження Big Data аналітики у фінансах, де важливими є керованість, прозорість обчислень і відтворюваність процедур [2, 3].

Середовище розгортання визначається потребою в горизонтальному масштабуванні та керованості ресурсів. Для цього застосовується кластерна модель, яка може бути реалізована як у локальному дата-центрі, так і у хмарній інфраструктурі. Хмарні обчислення характеризуються наданням ресурсів “на вимогу”, широким мережевим доступом та еластичним масштабуванням, що створює передумови для ефективної обробки великих фінансових масивів при змінному навантаженні. У межах модуля це означає можливість збільшення ресурсів під час масових регламентних розрахунків і зменшення у періоди низької активності.

Водночас хмарний контекст підсилює вимоги до безпеки й контролю доступу. Для фінансових даних необхідно передбачити криптографічний захист і аудит операцій. Як базовий стандарт шифрування розглядається AES відповідно до FIPS 197 [8], а також протоколювання доступу і критичних дій у системі [9]. Концептуально коректність узгодженого стану в розподіленому середовищі пов’язується з ідеями консенсусу, що обґрунтовуються протоколами на кшталт Paxos [10]. На практичному рівні це відображається у вимогах до версіонування наборів даних, контрольованих повторних запусків job-ів і фіксації метаданих для відтворюваності результатів [3, 10].

Архітектура модуля формується як набір компонентів із чітко визначеними функціями та інтерфейсами:

1. Ingestion-компонент. Приймає дані (файли/повідомлення), виконує первинну перевірку формату, фіксує метадані (джерело, час, тип набору) та розміщує дані у staging-зоні [3].

2. Компонент підготовки та контролю якості. Реалізує валідацію схеми, нормалізацію дат і валют, перевірку ключів, дедуплікацію, а також формує звіт про дефекти. Окремо передбачається виявлення пропусків і, за визначеними правилами, їх оброблення або імпутація, якщо це узгоджено з регламентом [11].

3. Storage Manager. Організує запис даних у HDFS (історичні, “холодні” набори) та у HBase (оперативні вибірки за ключем). Важливою умовою є коректне партиціювання і дизайн ключів, оскільки це впливає на рівномірність навантаження та ризик “гарячих точок”. В якості концептуальної основи вирівнювання розподілу ключів може використовуватися consistent hashing [6, 7].

4. Processing Orchestrator. Керує запуском Spark job-ів, передає параметри виконання (період, набір даних, перелік метрик), отримує статуси та зберігає результати у визначених форматах. Для підвищення продуктивності враховуються типові оптимізації Spark (розподіл партицій, мінімізація shuffle, використання кешування) [12].

5. Реєстр метаданих. Зберігає інформацію про версії датасетів, параметри запусків, результати перевірок якості, місця розміщення наборів у HDFS/HBase та технічні атрибути обробки. Такий реєстр є необхідним для data governance і відтворюваності розрахунків [2, 3].

6. Security/Audit-компонент. Забезпечує автентифікацію/авторизацію, розмежування прав доступу, журналювання та аудит критичних операцій, а також підтримку шифрування (за потреби) [8].

7. API-шар. Надає уніфікований інтерфейс для завантаження, запуску задач, отримання результатів, метрик і протоколів якості. Це підвищує керованість та спрощує інтеграцію зі

споживачами результатів [3].

Зазначені компоненти взаємодіють за принципом конвеєра: ingestion передає дані на підготовку, підготовка – в storage, storage забезпечує доступ обробці, Spark формує результати, а API надає результат і метадані зовнішнім системам. Такий підхід відповідає інженерним уявленням про проєктування розподілених сховищ і систем обробки великих даних, де важливою є декомпозиція на керовані компоненти [5].

Висновки. Запропонована архітектура модуля має трирівневу структуру. Рівень зберігання забезпечує роботу з великими історичними масивами та швидкий доступ до окремих записів. Рівень обробки відповідає за паралельні обчислення, агрегації, формування вітрин і перевірки якості. Прикладний рівень доступу забезпечує взаємодію із зовнішніми системами, запуск задач, отримання результатів і перегляд службової інформації.

Обґрунтовано доцільність використання HDFS для довготривалого розподіленого зберігання фінансових даних, HBase – для оперативного доступу за ключем, Apache Spark – для паралельної обробки великих масивів даних. Така комбінація дозволяє відокремити аналітичні й операційні навантаження та зменшити обмеження, характерні для централізованих сховищ.

Окрему увагу приділено питанням якості, безпеки та керованості даних. Передбачено валідацію, нормалізацію, дедуплікацію, оброблення пропусків, фіксацію метаданих, аудит дій і криптографічний захист. Це забезпечує прозорість обробки, можливість повторного виконання задач і підвищує надійність результатів фінансової аналітики.

Отже, запропонований підхід формує цілісний технологічний контур роботи з фінансовими даними: від їх приймання та підготовки до зберігання, паралельної обробки й надання результатів споживачам. Розроблена архітектура може бути використана як основа для створення масштабованих інформаційних систем фінансової аналітики, звітності та підтримки управлінських рішень.

Список літератури

1. Goldstein I., Spatt C. S., Ye M. Big data in finance. The Review of Financial Studies. 2021. Vol. 34, no. 7. P. 3213–3225.
2. Al-Afeef M., Ali O., Al-Tahat S. et al. The effect of big data governance on financial technology in Jordanian commercial banks: The mediation role of organizational culture. International Journal of Data and Network Science. 2023. Vol. 7, no. 3. P. 1283–1294.
3. Hajiheydari N., Delgosha M. S., Wang Y. et al. Exploring the paths to big data analytics implementation success in banking and financial service: an integrated approach. Industrial Management & Data Systems. 2021. Vol. 121, no. 12. P. 2498–2529.
4. Legault M. A practitioner's view on distributed storage systems: Overview, challenges and potential solutions. Technology Innovation Management Review. 2021. Vol. 11, no. 6. P. 32–41.
5. Zhang Xin. Distributed Storage and Parallel Processing Technology of Financial Big Data under Cloud Computing Platform. Procedia Computer Science. 2025. Vol. 262. P. 714–721.
6. Apache Software Foundation. Apache Hadoop: HDFS Architecture Guide [Електронний ресурс]. Режим доступу: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html (дата звернення: 24.01.2026).
7. Apache Software Foundation. Apache HBase® Reference Guide [Електронний ресурс]. Режим доступу: https://hbase.apache.org/book.html (дата звернення: 24.01.2026).
8. National Institute of Standards and Technology. FIPS PUB 197: Advanced Encryption Standard (AES). Gaithersburg, MD : NIST, 2001.
9. Regin R., Rajest S. S., Shynu T. A review of secure neural networks and big data mining applications in financial risk assessment. Central Asian Journal of Innovations on Tourism Management and Finance. 2023. Vol. 4, no. 2. P. 73–90.
10. Lamport L. Paxos Made Simple. ACM SIGACT News (Distributed Computing Column). 2001. Vol. 32, no. 4. P. 51–58.
11. Wang C., Shakhovska N., Sachenko A., Komar M. A New Approach for Missing Data Imputation in Big Data Interface. Information Technology and Control. – 2020. – Vol. 49. No 4. – Pp. 541-555.
12. Zaharia M., Chowdhury M., Franklin M. J., Shenker S., Stoica I. Spark: Cluster Computing with Working Sets. In: Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud'10). 2010. 10 p.

Пугінець А. Ю.[1], Зварич В. І.[2]
[1]студентка 4 курсу АСУ НУЛП, [2]асистент каф. АСУ НУ Львівська Політехніка
Науковий керівник к.х.н., асистент Зварич В. І., каф. АСУ НУ Львівська Політехніка

СМАРТ-СИСТЕМА ДЛЯ ДОГЛЯДУ ЗА КІМНАТНИМИ РОСЛИНАМИ

Вступ. Глобальний ринок кімнатних рослин оцінюється у 22.03 млрд доларів США з прогнозованим зростанням до 27.35 млрд доларів США у 2030 році [1]. Водночас 67% покупців рослин визнають себе «вбивцями рослин» через неправильний догляд, а 48% не впевнені у власній здатності утримати рослину живою [2]. Ключовою причиною є труднощі ідентифікації виду – без знання конкретних вимог рослини правильний догляд стає неможливим [3]. Це зумовлює потребу в автоматизованому інструменті, що поєднує ідентифікацію рослин та персоналізований план догляду.

Постановка задачі. Об'єкт дослідження – процес ідентифікації кімнатних рослин та планування догляду за ними з використанням методів комп'ютерного зору та машинного навчання.

Предмет дослідження. Вебсистема, що реалізує класифікацію зображень рослин, автоматичне формування персоналізованих планів догляду та мотивацію користувачів через ігрові механіки.

Мета дослідження. Розробка вебсистеми PlantCare, що інтегрує модуль автоматичної ідентифікації видів рослин на основі згорткової нейронної мережі EfficientNet-B4, адаптивний калькулятор параметрів догляду та підсистему гейміфікації.

Основний матеріал. Система PlantCare реалізує комплексний підхід до автоматизованого догляду за кімнатними рослинами. Архітектура побудована на основі клієнт-серверної моделі: серверна частина (FastAPI, Python) реалізує REST API з 26 ендпоінтами та зберігає дані у PostgreSQL; клієнтська частина (Angular 17, TypeScript) забезпечує SPA-інтерфейс з адаптивністю від 360 px [6, 7].

Модуль розпізнавання базується на архітектурі EfficientNet-B4 із кастомною класифікаційною головою. Вибір архітектури обґрунтовано оптимальним балансом точності (82.9% на ImageNet), кількості параметрів (19.3 M) та придатності для CPU-інференсу [4]. Для підвищення якості застосовано двофазний Transfer Learning, Data Augmentation (9 трансформацій) [5], Mixup ($\alpha = 0.2$) [8], Label Smoothing ($\varepsilon = 0.1$) [9] та Test-Time Augmentation (5 трансформацій). На тестовій вибірці з 1501 зображення (57 видів) досягнуто Top-1 Accuracy 92.67%.

Адаптивний алгоритм розрахунку об'єму поливу враховує чотири фактори: вид рослини (коефіцієнт водопотреби), розмір горщика (об'єм за формулою циліндра), сезон (коефіцієнти 0.5-1.0) та географічне розташування з інверсією для південної півкулі. Підхід базується на моделях оптимального зрошення [10] та адаптований для кімнатних умов без IoT-датчиків.

Підсистема гейміфікації нараховує XP за виконання дій догляду (10-30 очок залежно від типу дії) з модифікаторами вчасності та стріку, підтримує 5 рівнів розвитку та 5 типів бейджів. Згідно з дослідженнями, ігрові механіки позитивно впливають на залученість користувачів [11] та підвищують регулярність виконання завдань у екологічних застосунках до 40% [12].

У таблиці 1 наведено порівняння системи PlantCare з існуючими аналогами.

Таблиця 1 - Порівняння системи PlantCare з аналогами

Критерій	**PlantNet**	**PictureThis**	**Planta**	**Greg**	**PlantCare**
Ідентифікація за фото	+	+	+	–	+
Адаптивний об'єм поливу	–	–	–	–	+

Продовження Таблиці 1

Горщик + сезон + геолокація	–	Частково	Частково	Частково	+
Гейміфікація (XP, рівні, бейджі)	–	–	–	–	+
Веб-інтерфейс	+	–	–	–	+
Відкритий код	Частково	–	–	–	+

Функціональне тестування підтвердило виконання 15/15 функціональних та 9/9 нефункціональних вимог (100%). Час інференсу на CPU складає 2000-3500 мс при нормативі ≤ 5000 мс, завантаження Dashboard ~ 1200 мс при нормативі ≤ 3000 мс, що відповідає стандартам якості програмного забезпечення [13].

Висновки. У рамках дослідження розроблено веб-систему PlantCare, яка поєднує автоматичну ідентифікацію виду рослини за фотографією (Top-1 Accuracy 92.67%), персоналізований план догляду з точним розрахунком об'єму поливу в мілілітрах та систему гейміфікації для підтримки регулярності догляду. Порівняльний аналіз показав, що жоден з існуючих аналогів не реалізує повного поєднання ідентифікації, адаптивного розрахунку догляду та гейміфікації в єдиній системі. Подальший розвиток може включати інтеграцію IoT-датчиків вологості ґрунту та освітленості, а також розширення каталогу рослин за рахунок відкритих ботанічних датасетів.

Список літератури

1. Mordor Intelligence. Indoor Plant Market Size & Share Outlook to 2030. *Mordor Intelligence*. 2025. https://www.mordorintelligence.com/industry-reports/indoor-plant-market (Дата звернення: 10.05.2026).
2. Del Castillo, I. This Is How Many Houseplants the Average Plant Parent Has Killed. *Apartment Therapy*, May 11, 2022. https://www.apartmenttherapy.com/dead-houseplants-average-37076429 (Дата звернення: 11.05.2026).
3. Wäldchen J., Mäder P. Machine learning for image-based species identification. *Methods in Ecology and Evolution*. 2018. Vol. 9, No. 11. P. 2216–2225.
4. Tan M., Le Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th ICML*. 2019. P. 6105–6114.
5. Shorten C., Khoshgoftaar T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*. 2019. Vol. 6, No. 60. P. 1–48.
6. FastAPI Documentation. URL: https://fastapi.tiangolo.com/
7. Angular Documentation, Version 17. URL: https://angular.dev/
8. Zhang H., Cissé M., Dauphin Y. N., Lopez-Paz D. mixup: Beyond Empirical Risk Minimization. *Proceedings of ICLR*. 2018.
9. Müller R., Kornblith S., Hinton G. When Does Label Smoothing Help? *NeurIPS*. 2019. Vol. 32. P. 4694–4703.
10. Fereres E., Soriano M. A. Deficit irrigation for reducing agricultural water use. *Journal of Experimental Botany*. 2007. Vol. 58, No. 2. P. 147–159.
11. Hamari J., Koivisto J., Sarsa H. Does Gamification Work? *Proceedings of the 47th HICSS*. IEEE, 2014. P. 3025–3034.
12. Pratama A. R., Hidayat R. Gamification for Sustainable Behavior: A Systematic Literature Review. *Sustainability*. 2023. Vol. 15, No. 3. 108529.
13. ДСТУ ISO/IEC 25010:2025 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Модель якості продукту (ISO/IEC 25010:2023, IDT).

Вишинська Н.М.
студентка 3 курсу НУ «Львівська політехніка»
Науковий керівник д.т.н., професор Теслюк В.М., кафедра АСУ НУ «Львівська політехніка»

ІНФОРМАЦІЙНА СИСТЕМА РЕКОМЕНДАЦІЙ ЗАКЛАДІВ ХАРЧУВАННЯ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Вступ. Сфера громадського харчування в Україні є одним із найбільш динамічно зростаючих секторів економіки. За даними Державної податкової служби, у 2025 році задекларована виручка галузі перевищила 246 млрд грн - на 21,7% більше порівняно з 2024 роком, а загальна кількість суб'єктів господарювання перевищує 43 тисячі [1]. Зростання ринку супроводжується збільшенням обсягу інформації, яку споживач змушений опрацьовувати при виборі закладу: відгуки, рейтинги, меню, ціни, розташування, режим роботи.

Наявні рішення (Google Maps, TripAdvisor та подібні платформи) реалізують переважно базову фільтрацію за наперед визначеними параметрами. Вони не враховують індивідуальний профіль користувача, його попередній досвід та приховані вподобання. У роботі досліджено підходи до побудови персоналізованої рекомендаційної системи закладів харчування та виконано порівняльний аналіз алгоритмів машинного навчання для вибору оптимального рішення.

Постановка задачі.

Об'єктом дослідження є процес персоналізованого вибору закладів харчування користувачем в умовах великої кількості альтернатив. Предметом дослідження є методи та алгоритми машинного навчання для побудови рекомендаційних систем у сфері громадського харчування. Метою роботи є розробка інформаційної системи рекомендацій закладів харчування на основі гібридного підходу з використанням методів машинного навчання, що забезпечує персоналізацію та стійкість до проблеми холодного старту.

Задача відноситься до класу задач фільтрації та ранжування інформації. Нехай задано множину користувачів $U = \{u_1, u_2, ..., u_n\}$, множину закладів харчування $R = \{r_1, r_2, ..., r_m\}$ та множину контекстних параметрів $C = \{c_1, c_2, ..., c_k\}$. Для кожного користувача u_i та заданого контексту c_j необхідно побудувати функцію релевантності $f(u_i, r_l, c_j)$, яка відображає ступінь відповідності закладу r_l запиту користувача, та сформувати впорядкований список топ-N закладів з найвищими значеннями f.

Основні вимоги до системи: персоналізація рекомендацій на основі профілю користувача; урахування контекстних факторів (бюджет, склад компанії, час доби); механізм роботи з новими користувачами (проблема холодного старту); час відповіді не більше 3–5 секунд.

Основний матеріал. Відповідно до класифікації, наведеної в роботі [2], усі підходи до побудови рекомендаційних систем поділяються на три основні категорії. Контентна фільтрація (CBF) базується на аналізі атрибутів об'єктів та профілю користувача; її перевагою є незалежність від інших користувачів, недоліком є необхідність детального опису характеристик об'єктів. Колаборативна фільтрація (CF) будує матрицю «користувач–об'єкт» на основі оцінок і знаходить схожих користувачів; вона ефективна для контенту, який важко описати метаданими, але погано працює для нових акаунтів (проблема холодного старту). Гібридна фільтрація поєднує обидва підходи, компенсуючи їхні слабкі сторони [2].

У контексті рекомендаційних систем закладів харчування (Food Recommendation Systems, FRS) виділяють чотири типи залежно від підходу до формування рекомендацій: за смаковими вподобаннями, за нутриційними потребами, за комбінацією обох критеріїв та групові рекомендаційні системи [3]. Для даної задачі обрано гібридний підхід, що поєднує колаборативну та контентну фільтрацію, оскільки він забезпечує найкращий баланс між персоналізацією та стійкістю до холодного старту. Порівняння розглянутих алгоритмів наведено в таблиці 1.

Таблиця 1. Порівняння алгоритмів рекомендацій

Алгоритм	Холодний старт	Облік контексту	Масштабованість
User-based CF	Погано	Ні	Низька
Item-based CF	Задовільно	Ні	Середня
SVD	Погано	Частково	Висока
Content-based	Добре	Частково	Висока
SVD (гібридна)	Добре	Так	Висока

Як основу моделі обрано алгоритм SVD із бібліотеки Surprise, що реалізує матричну факторизацію. Алгоритм добре справляється з розрідженими даними та забезпечує високу масштабованість [4]. Для нових користувачів, у яких ще немає оцінок, передбачено резервний механізм на основі популярності закладів.

Для навчання та тестування моделі використано Yelp Academic Dataset публічний набір даних, що містить понад 200 000 відгуків про ресторани США. Препроцесинг включав фільтрацію користувачів з менш ніж 5 відгуками та ресторанів з менш ніж 10 оцінками. Реалізацію виконано мовою Python з використанням бібліотек Surprise, pandas, numpy та scikit-learn. Якість моделі оцінювалась за метриками RMSE, MAE, Precision@K та Recall@K (поріг релевантності - оцінка $\geq 4{,}0$). Результати порівняння алгоритмів наведено в таблиці 2.

Таблиця 2. Порівняння алгоритмів за метриками якості

Алгоритм	RMSE	MAE	Precision@10	Recall@10
NormalPredictor (base)	1,5230	1,2180	0,0210	0,0085
KNNBasic (user-based)	1,0420	0,8190	0,0580	0,0340
KNNBasic (item-based)	1,0180	0,8010	0,0610	0,0370
SVD	0,9140	0,7180	0,0820	0,0510
SVD (гібридна)	0,8870	0,6950	0,0940	0,0590

Гібридний SVD показав найкращі результати серед усіх розглянутих алгоритмів: RMSE знижено з 1,523 (базовий NormalPredictor) до 0,887, що становить покращення на 41,7%. Precision@10 зросла з 0,021 до 0,094 - у 4,5 рази. Це підтверджує доцільність гібридного підходу для задачі персоналізованих рекомендацій закладів харчування.

Висновки. У роботі виконано порівняльний аналіз алгоритмів побудови рекомендаційних систем у контексті задачі персоналізованого вибору закладів харчування. Розглянуто підходи контентної, колаборативної та гібридної фільтрації; встановлено, що гібридна модель на основі SVD забезпечує найкращі показники за всіма метриками: RMSE = 0,887, Precision@10 = 0,094. Отримані результати свідчать про ефективність матричної факторизації для виявлення прихованих вподобань користувачів у поєднанні з контентною складовою для подолання проблеми холодного старту. Подальші дослідження передбачають апробацію граф-орієнтованих методів та підтримку групових рекомендацій із застосуванням стратегії Least Misery.

Список літератури

1. Державна податкова служба України. Статистичні дані щодо ресторанного бізнесу за 2025 рік. URL: https://tax.gov.ua/.

2. Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems Handbook. Springer, 2011. 842 p.

3. Trang Tran T.N., Atas M., Felfernig A., Stettinger M. An overview of recommender systems in the healthy food domain. Journal of Intelligent Information Systems. 2018. Vol. 50. P. 501–526.

4. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. Computer. 2009. Vol. 42, No. 8. P. 30–37.

5. Isinkaye F.O., Folajimi Y.O., Ojokoh B.A. Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal. 2022. Vol. 16, No. 3. P. 261–273.

Тарбай Ю.О.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ОЦІНКИ ФІНАНСОВИХ РИЗИКІВ ПІДПРИЄМСТВА НА ОСНОВІ ТЕХНОЛОГІЇ АНАЛІЗУ ВЕЛИКИХ ДАНИХ

Вступ. Сучасні умови функціонування підприємств характеризуються зростанням рівня невизначеності та посиленням впливу внутрішніх і зовнішніх факторів, що безпосередньо відображається на фінансовій стійкості суб'єктів господарювання. Коливання ринкової кон'юнктури, нестабільність макроекономічного середовища, зростання конкуренції, а також наслідки глобальних кризових явищ і воєнних ризиків зумовлюють необхідність удосконалення підходів до оцінки та управління фінансовими ризиками підприємств [1, 2].

Постановка задачі. Результати аналізу фінансових ризиків підприємства, сучасних підходів до їх оцінювання та можливостей систем підтримки прийняття рішень свідчать про необхідність удосконалення інструментів фінансового ризик-менеджменту. Традиційні методи аналізу, попри їх поширеність, не забезпечують достатньої точності та оперативності в умовах зростання обсягів фінансових даних і підвищення динамічності зовнішнього середовища [3]. Це зумовлює доцільність використання інтелектуальних підходів, заснованих на аналізі великих даних і методах машинного навчання [4].

У зв'язку з цим у межах даного дослідження ставиться задача розроблення системи підтримки прийняття рішень для оцінки фінансових ризиків підприємства, яка поєднує можливості сучасних аналітичних методів із засобами автоматизованої обробки фінансово-економічної інформації. Така система має бути орієнтована на практичне використання у фінансовому управлінні та забезпечувати формування обґрунтованих рекомендацій для прийняття управлінських рішень.

Об'єктом дослідження є процес управління фінансовими ризиками підприємства. Предметом дослідження є методи, моделі та програмні засоби підтримки прийняття рішень для оцінки фінансових ризиків на основі аналізу великих даних. Метою дослідження є проєктування системи підтримки прийняття рішень для оцінки фінансових ризиків підприємства з використанням методів аналізу великих даних та машинного навчання.

Основний матеріал. Запропонована архітектура системи підтримки прийняття рішень базується на модульному підході, що дозволяє відокремити основні функціональні компоненти системи та забезпечити їх незалежний розвиток і модифікацію. Такий підхід є доцільним у контексті оцінки фінансових ризиків, оскільки дає змогу адаптувати систему до змін вхідних даних і використовуваних аналітичних моделей без суттєвого втручання в загальну структуру системи.

Загальну архітектуру системи підтримки прийняття рішень для оцінки фінансових ризиків підприємства подано на рисунку 1.

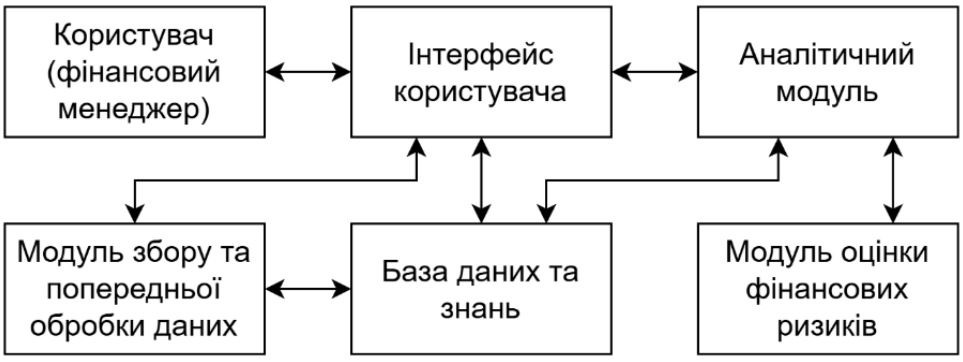


Рисунок 1 – Архітектура системи підтримки прийняття рішень для оцінки фінансових ризиків підприємства

Як видно з рисунку 1, система складається з кількох взаємопов'язаних модулів, кожен з яких виконує визначені функції в процесі підтримки управлінських рішень.

Центральним елементом архітектури є аналітичний модуль, у якому реалізуються методи машинного навчання для оцінювання фінансових ризиків. У межах даної роботи як базовий інструмент аналітичного аналізу розглядається метод випадкового лісу, що забезпечує високу точність класифікації фінансових станів підприємства та стійкість до впливу шуму у вхідних даних [4-6]. Аналітичний модуль отримує підготовлені дані з модуля збору та попередньої обробки і формує результати оцінки рівня фінансових ризиків.

Використання методу випадкового лісу для оцінювання фінансових ризиків підприємства є доцільним з кількох причин. По-перше, алгоритм добре працює з наборами даних, що містять різнорідні фінансові показники та не вимагає жорстких припущень щодо їх розподілу. По-друге, метод є стійким до перенавчання за рахунок використання випадкового відбору ознак і навчальних підвибірок. По-третє, випадковий ліс забезпечує можливість оцінювання важливості окремих ознак, що підвищує інтерпретованість результатів аналізу в контексті фінансового управління [5, 6].

Вхідною інформацією для аналітичного модуля є вектор ознак, сформований на основі фінансових показників підприємства після етапу попередньої обробки даних. Кожен вектор ознак відповідає окремому звітному періоду та містить значення показників ліквідності, фінансової стійкості, рентабельності та ділової активності. На основі цих даних модель машинного навчання здійснює класифікацію фінансового стану підприємства за рівнями фінансового ризику.

Процес функціонування аналітичного модуля передбачає два основні етапи: навчання моделі та її застосування для оцінювання фінансових ризиків. На етапі навчання використовується історичний набір даних, у якому для кожного спостереження відомий відповідний рівень фінансового ризику. Модель Random Forest навчається на основі множини навчальних прикладів шляхом побудови окремих дерев рішень та формування ансамблевої моделі. На етапі застосування навчена модель використовується для оцінювання фінансових ризиків за новими або оновленими даними.

Модуль збору та попередньої обробки даних призначений для агрегування фінансово-економічної інформації з різних джерел, її очищення, нормалізації та підготовки до подальшого аналізу. Реалізація цього модуля є необхідною умовою коректної роботи аналітичного блоку, оскільки якість вхідних даних безпосередньо впливає на достовірність результатів оцінювання фінансових ризиків [3].

Важливу роль у запропонованій архітектурі відіграє база даних та знань, яка забезпечує зберігання фінансових показників, результатів аналізу, а також історичних даних, необхідних для навчання та перевірки моделей машинного навчання. Наявність такої бази дозволяє реалізувати механізми накопичення знань і підвищувати адаптивність системи до змін фінансового середовища.

Модуль оцінки фінансових ризиків відповідає за інтерпретацію результатів аналітичного аналізу та формування інтегральної оцінки рівня ризику. У межах цього модуля здійснюється перетворення вихідних результатів моделей машинного навчання у форму, зручну для сприйняття та використання в процесі прийняття управлінських рішень.

Взаємодія користувача з системою забезпечується за допомогою інтерфейсу користувача, який виконує функції візуалізації результатів аналізу, представлення рекомендацій та підтримки процесу прийняття рішень. Через інтерфейс фінансовий менеджер отримує доступ до інформації про рівень фінансових ризиків, а також може аналізувати альтернативні управлінські сценарії. Логіка інформаційних потоків у системі підтримки прийняття рішень для оцінки фінансових ризиків підприємства побудована відповідно до послідовності виконання основних функціональних операцій та відображає рух даних і результатів аналізу між структурними модулями системи. На початковому етапі взаємодії користувач системи (фінансовий менеджер) через інтерфейс користувача ініціює процес оцінки фінансових ризиків шляхом формування запиту, вибору періоду аналізу та визначення набору фінансових показників. Даний інформаційний потік спрямовується від користувача до інтерфейсу

користувача, який виконує функцію приймання та попередньої інтерпретації запиту. Після цього сформований запит передається з інтерфейсу користувача до модуля збору та попередньої обробки даних. У межах цього модуля здійснюється збирання фінансово-економічної інформації з визначених джерел, її очищення від пропусків і аномальних значень, а також нормалізація та приведення до формату, придатного для подальшого аналізу.

Результати попередньої обробки даних надходять до бази даних та знань, де здійснюється їх збереження та накопичення. База даних забезпечує централізоване зберігання як поточних фінансових показників, так і історичних даних, що використовуються для навчання та перевірки моделей машинного навчання. Далі підготовлені дані з бази даних передаються до аналітичного модуля, у якому реалізуються алгоритми машинного навчання, зокрема метод випадкового лісу. В аналітичному модулі здійснюється обробка вхідного вектора ознак, формування прогнозних оцінок та виявлення рівня фінансових ризиків підприємства на основі багатовимірного аналізу даних. Отримані результати аналітичного аналізу передаються до модуля оцінки фінансових ризиків, який виконує інтерпретацію вихідних даних моделей машинного навчання та формує інтегральну оцінку рівня фінансового ризику. У цьому модулі результати аналізу трансформуються у форму, зручну для використання в процесі прийняття управлінських рішень.

На завершальному етапі результати оцінювання фінансових ризиків та сформовані рекомендації передаються до інтерфейсу користувача, де вони відображаються у вигляді аналітичних звітів, графічних візуалізацій або рекомендацій щодо подальших управлінських дій. Таким чином забезпечується зворотний зв’язок між аналітичною частиною системи та користувачем. Запропонована логіка потоків забезпечує послідовний і цілісний процес оцінювання фінансових ризиків підприємства, поєднуючи етапи збору даних, аналітичної обробки та підтримки прийняття управлінських рішень у межах єдиної системи.

Висновки. Проведене дослідження дозволило узагальнити підходи до оцінки фінансових ризиків підприємства в умовах зростання невизначеності та обсягів даних. Встановлено, що традиційні методи аналізу не забезпечують достатньої точності та адаптивності, що обґрунтовує доцільність використання методів машинного навчання. Обґрунтовано застосування систем підтримки прийняття рішень як ефективного інструменту фінансового ризик-менеджменту. Запропоновано архітектуру системи, побудовану за модульним принципом, що забезпечує гнучкість, масштабованість і можливість інтеграції аналітичних моделей. Ключовим елементом системи визначено аналітичний модуль на основі методу випадкового лісу, який забезпечує ефективну обробку багатовимірних фінансових даних, стійкість до шуму та інтерпретованість результатів. Встановлено, що якість оцінювання фінансових ризиків залежить від ефективності підготовки даних, зокрема процедур очищення, нормалізації та формування вектора ознак. Розроблена логіка інформаційних потоків забезпечує узгоджену взаємодію між компонентами системи та підтримку прийняття управлінських рішень.

Список літератури

1. Ku C., Morriss A. International financial centers as a model: Facilitating growth and development by connecting to international legal frameworks. Law and Development Review. 2021. Vol. 14(2). P. 429–464.

2. Дашко Т. В., Ковальчук А. А. Використання штучного інтелекту у фінансах. Економіка і прогнозування. 2022. №3. С. 75–82.

3. Литовченко О.Ю. Підходи до ідентифікації та оцінки фінансових ризиків підприємства. Економіка та суспільство. 2018. № 16. С. 398–404.

4. Breiman L. Random forests. Machine Learning. 2001. Vol. 45(1). P. 5–32.

5. Du G., Elston F. Financial risk assessment to improve the accuracy of financial prediction in the internet financial industry using data analytics models. Operations Management Research. 2022. Vol. 15(3). P. 925–940.

6. Uddin M.S., Chi G., Al Janabi M.A.M., et al. Leveraging random forest in micro-enterprises credit risk modelling for accuracy and interpretability. International Journal of Finance & Economics. 2022. Vol. 27(3). P. 3713–3729.

Григорян Д. М.
студентка 4 курсу кафедри АСУ, ІКНІ, НУ Львівська Політехніка
Науковий керівник к.т.н., доцент кафедри АСУ Стрямець С. П.

МЕТОДИ ТА ЗАСОБИ КЛАСИФІКАЦІЇ АКНЕ НА ОСНОВІ МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ

Вступ. Акне (вугрова хвороба) є одним із найпоширеніших хронічних захворювань шкіри у світі – за даними Всесвітньої організації охорони здоров'я, воно уражає 9,4% населення та займає восьме місце серед найпоширеніших шкірних захворювань [1]. Традиційна діагностика здійснюється дерматологом під час очного огляду, проте доступ до вузькопрофільних спеціалістів залишається нерівномірним: у великих містах черги тривають тижнями, а в малих населених пунктах такі фахівці можуть бути взагалі відсутні. Розвиток методів глибокого навчання відкрив нові можливості для автоматизованої класифікації медичних зображень, однак більшість існуючих підходів вимагають сотень або тисяч анотованих зображень на клас, збір яких є складним і дорогим [2]. Окремою проблемою є те, що наявні системи повертають лише числовий клас, не надаючи жодних практичних рекомендацій для кінцевого користувача. Саме ці прогалини – висока залежність класичних підходів від великих датасетів та відсутність персоналізованих рекомендацій – і стали відправною точкою для даної роботи, що доводить її актуальність.

Постановка задачі. Метою роботи є дослідження методів та засобів класифікації ступеня акне за зображенням обличчя і розроблення системи з генерацією персоналізованих рекомендацій щодо догляду за шкірою на основі методів комп'ютерного зору та великих мовних моделей. Об'єктом дослідження є процеси автоматизованої класифікації медичних зображень та генерації персоналізованих рекомендацій в умовах обмежених навчальних даних. Предметом дослідження є методи few-shot learning та трансферного навчання для класифікації зображень, а також підходи до побудови RAG-систем для генерації медично орієнтованих рекомендацій.

Основний матеріал. Для вирішення задачі класифікації зображень в умовах обмежених навчальних даних досліджувались дві стратегії. Стратегія трансферного навчання (Transfer Learning) потребує велику кількість зображень на клас і здійснює пряму класифікацію через дотренування попередньо навченої мережі. Стратегія навчання з малою кількістю прикладів (Few-Shot Learning) є принципово іншою парадигмою: модель навчається не класифікувати безпосередньо, а порівнювати зображення між собою, знаходячи схожість із відомими прикладами. Це дозволяє досягати прийнятної якості за наявності лише 10–20 прикладів на клас [3].

На першому етапі проводились експерименти з трансферним навчанням на публічному датасеті ACNE04 (1378 зображень, 4 класи тяжкості). Досліджувались архітектури ResNet18 та EfficientNet-B0 із такими техніками, як зважена крос-ентропія для компенсації незбалансованості класів, Label Smoothing (α=0,05) та Mixup-аугментація (α=0,4) [4]. Навчання проводилось у два етапи fine-tuning: спочатку заморожувались всі шари, крім класифікаційної голівки, потім поступово розморожувались верхні блоки енкодера.

На другому етапі реалізовано протипові мережі (Prototypical Networks) з енкодером ResNet34 та projection head для few-shot класифікації п'яти рівнів тяжкості акне: від класу «чиста шкіра» (acne0) до важкого ступеня (acne4). Вхідний вектор зображення розмірністю 512, отриманий через Global Average Pooling, проєктується projection head у простір ознак розмірністю 128 та нормалізується за L2-нормою. Прототип кожного класу обчислюється як нормалізоване середнє embedding опорних прикладів support set. Класифікація нового зображення виконується через косинусну схожість з прототипами та softmax із temperature scaling (T=10) [3]. Навчання організовано через 5-way 5-shot епізоди на власноруч відібраній вибірці із 22 зображень на клас.

Порівняльні результати усіх досліджених підходів наведено у таблиці 1.

Таблиця 1 - Порівняльні результати стратегій навчання моделі класифікації

Стратегія	Архітектура	Конфігурація	Val Accuracy	Std
Transfer Learning	ResNet18	ACNE04, 4 класи	75.45%	–
Transfer Learning	EfficientNet-B0	ACNE04, 4 класи	77.62%	–
Few-Shot Learning	ResNet34	4-way 5-shot	93,75%	**±0,036**
Few-Shot Learning	ResNet34	5-way 5-shot	**87,25%**	**±0,054**

Val Accuracy (точність на валідаційній вибірці) – основна метрика оцінки моделі, що визначає частку правильно класифікованих зображень серед усіх зразків валідаційного набору. У контексті few-shot learning вона обчислюється як середнє значення episode accuracy по фіксованій кількості валідаційних епізодів: на кожному епізоді випадково формується support set та query set, після чого підраховується частка правильно визначених класів серед query-зображень. Додатково відстежується стандартне відхилення (std) як показник стабільності моделі між епізодами.

Для генерації персоналізованих рекомендацій реалізовано RAG-модуль на основі локальної LLM Gemma-3-4B [5]. Модуль виконує пошук релевантних фрагментів дерматологічної бази знань за ключовими словами та формує структурований промпт, що містить результат класифікації та відповіді 17-параметрової анкети користувача (тип шкіри, гормональні фактори, харчові звички, спосіб життя). Використання локальної моделі є принциповим архітектурним рішенням: медичні дані не передаються до зовнішніх хмарних сервісів.

Серверна частина реалізована на FastAPI (Python 3.12) з JWT-автентифікацією та збереженням зображень у хмарному сховищі AWS S3. Структуровані дані (користувачі, результати класифікації, анкети та рекомендації) зберігаються у реляційній базі даних MSSQL.

На рисунку 1 представлено діаграму варіантів використання системи класифікації акне, яка відображає функціональні вимоги до розроблюваної системи.

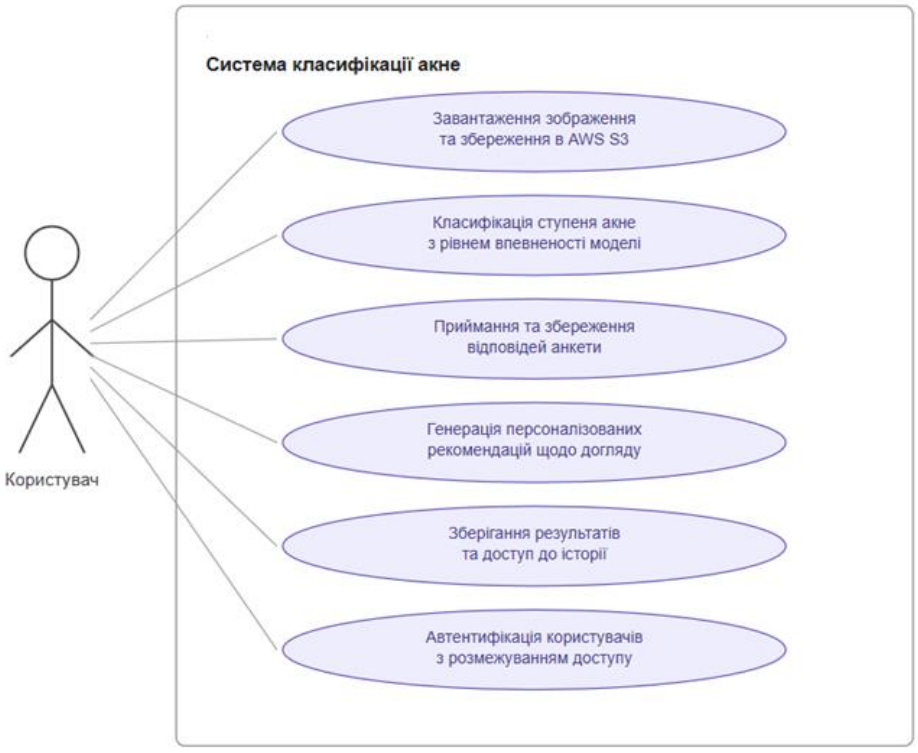


Рис. 1. Use Case діаграма системи класифікації акне

Діаграма містить єдиного актора – користувача, який взаємодіє із шістьма варіантами використання: завантаження зображення та збереження в AWS S3, класифікація ступеня акне з

рівнем впевненості моделі, приймання та збереження відповідей анкети, генерація персоналізованих рекомендацій щодо догляду, зберігання результатів та доступ до історії, а також автентифікація користувачів з розмежуванням доступу. Кожен варіант використання відповідає окремій функціональній вимозі системи та реалізує конкретну цінність для кінцевого користувача.

Висновки. Досліджено методи класифікації ступеня акне за зображенням обличчя та розроблено систему з генерацією персоналізованих рекомендацій. Порівняльний аналіз підходів показав, що трансферне навчання (ResNet18, EfficientNet-B0) на датасеті ACNE04 є обмеженим при малих вибірках. Найкращі метрики забезпечив підхід few-shot learning на основі протипових мереж з енкодером ResNet34: при конфігурації 4-way 5-shot досягнуто точності 93,75% (std ±0,036), при розширенні до п'яти класів у конфігурації 5-way 5-shot – 87,25% (std ±0,054), що перевищує визначений поріг прийнятності 85%. Інтеграція класифікатора з RAG-модулем на базі LLM Gemma-3-4B дозволяє формувати структуровані рекомендації українською мовою з урахуванням індивідуального профілю користувача, що підтверджує практичну цінність розробленої системи.

Список літератури

1. WHO Global Health Estimates 2021. URL: https://www.who.int

2. Watanabe K. et al. Deep Learning-Based Acne Severity Classification Using Standardized Facial Images // Cureus. – 2025. – Vol. 17. DOI: 10.7759/cureus.91944

3. Snell J., Swersky K., Zemel R. Prototypical Networks for Few-shot Learning // NeurIPS. – 2017. URL: https://arxiv.org/abs/1703.05175

4. Zhang H. et al. mixup: Beyond Empirical Risk Minimization // ICLR. – 2018. URL: https://arxiv.org/abs/1710.09412

5. Lewis P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // NeurIPS. – 2020. URL: https://arxiv.org/abs/2005.11401

Василиків В.Д.
студентка 4 курсу кафедри АСУ, ІКНІ, НУ Львівська Політехніка
Науковий керівник асистентка кафедри АСУ Бадзь В.М., ІКНІ, НУ Львівська Політехніка

ІНФОРМАЦІЙНА СИСТЕМА ВИЗНАЧЕННЯ ФІЗИЧНИХ ХАРАКТЕРИСТИК ТВАРИН З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ

Вступ. Сучасний розвиток технологій комп'ютерного зору та штучного інтелекту сприяє автоматизації процесів у сільському господарстві та тваринництві. Одним із перспективних напрямів є автоматизоване визначення фізичних характеристик тварин за цифровими зображеннями, що дозволяє зменшити потребу у ручних вимірюваннях та спеціалізованому обладнанні. Подібні системи можуть використовуватися для оцінювання маси тіла, параметрів росту та контролю фізичного стану тварин у межах концепції Smart Farming.

Методи комп'ютерного зору та машинного навчання вже застосовуються для автоматизованого аналізу фотографій великої рогатої худоби та прогнозування її фізичних параметрів[1]. Використання сегментації зображень дає можливість виділити область тіла тварини та зменшити вплив фону на результати прогнозування[2]. Для реалізації таких систем часто використовуються звичайні RGB-зображення, що дозволяє уникнути застосування дорогих сенсорів або спеціалізованого обладнання[3]. Водночас більшість доступних відкритих наборів даних орієнтовані саме на задачі оцінювання ваги великої рогатої худоби, що зумовило вибір цього напряму для практичної реалізації системи.

Постановка задачі. Метою роботи є підвищення ефективності визначення ваги тварини шляхом розроблення програмної системи для автоматизованого оцінювання ваги корови за фотографією із використанням методів комп'ютерного зору та машинного навчання.

Розроблена система має забезпечити швидке та зручне отримання результатів без необхідності використання спеціалізованого вагового обладнання, що сприятиме спрощенню процесу контролю стану тварин, зменшенню витрат часу та підвищенню ефективності ведення фермерського господарства.

Об’єктом дослідження є процеси автоматизованого аналізу зображень великої рогатої худоби для оцінювання фізичних параметрів тварин.

Предметом дослідження є методи та засоби комп’ютерного зору, глибокого навчання, сегментації зображень та регресійного аналізу для прогнозування ваги тварини за фотографією.

Для досягнення поставленої мети необхідно проаналізувати сучасні підходи до автоматизованого аналізу зображень тварин, дослідити методи сегментації та регресійного прогнозування, розробити архітектуру програмної системи та реалізувати модель машинного навчання для оцінювання фізичних параметрів тварини за цифровим зображенням.

Основний матеріал. Розроблена інформаційна система базується на використанні методів комп’ютерного зору та глибокого навчання для автоматизованого аналізу зображень тварин. Архітектура системи реалізована за модульним принципом і включає етапи попередньої обробки зображення, класифікації об’єкта, сегментації області тіла тварини та подальшого прогнозування фізичних характеристик.

Для перевірки наявності тварини на зображенні використано згорткову нейронну мережу ResNet18, яка забезпечує бінарну класифікацію вхідних даних. Після підтвердження належності об’єкта до цільового класу застосовується модель сегментації U-Net, що дозволяє виділити область тіла тварини та мінімізувати вплив фону на якість подальшого аналізу.

Основний модуль системи реалізовано у вигляді двопотокової регресійної моделі, структурну схему якої наведено на рисунку 1. Перший потік виконує аналіз RGB-зображення, а другий – аналіз сегментаційної маски, сформованої після виділення тварини. Об’єднання ознак із двох потоків дозволяє підвищити точність прогнозування фізичних параметрів за рахунок одночасного врахування текстурних та геометричних характеристик об’єкта.

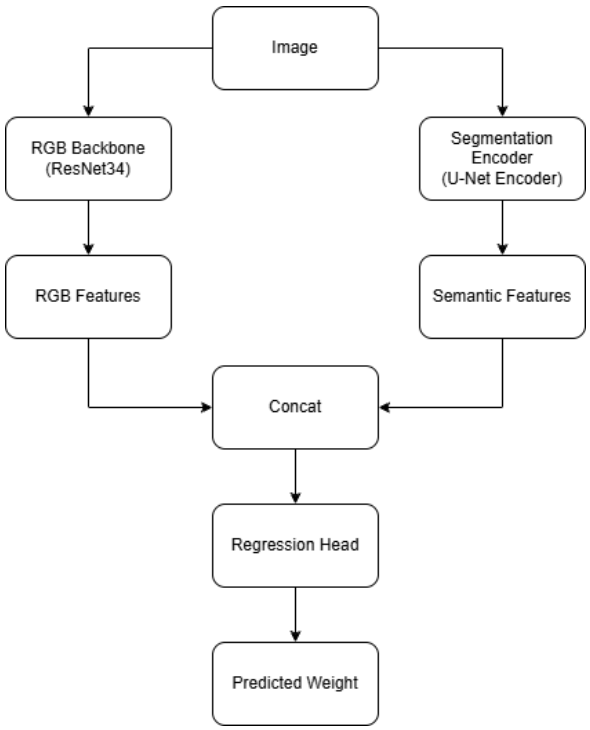


Рисунок 1. Структурна схема двопотокової моделі

Для реалізації та навчання моделей використано бібліотеку PyTorch та метод оптимізації Adam. Якість роботи системи оцінювалася за допомогою метрик MAE, RMSE, MAPE та коефіцієнта детермінації R^2, що дозволило комплексно оцінити точність регресійного прогнозування.

У межах практичної апробації система застосовується для прогнозування ваги великої рогатої худоби за цифровими зображеннями. Приклад результату роботи моделі та інтерфейсу прогнозування наведено на рисунку 2. Вибір саме цього сценарію дослідження обумовлений доступністю відкритих наборів даних із фотографіями корів та відповідними значеннями маси тіла. Водночас запропонована архітектура системи є масштабованою та може бути адаптована для визначення інших фізичних характеристик і роботи з різними видами тварин шляхом донавчання моделей на відповідних наборах даних.

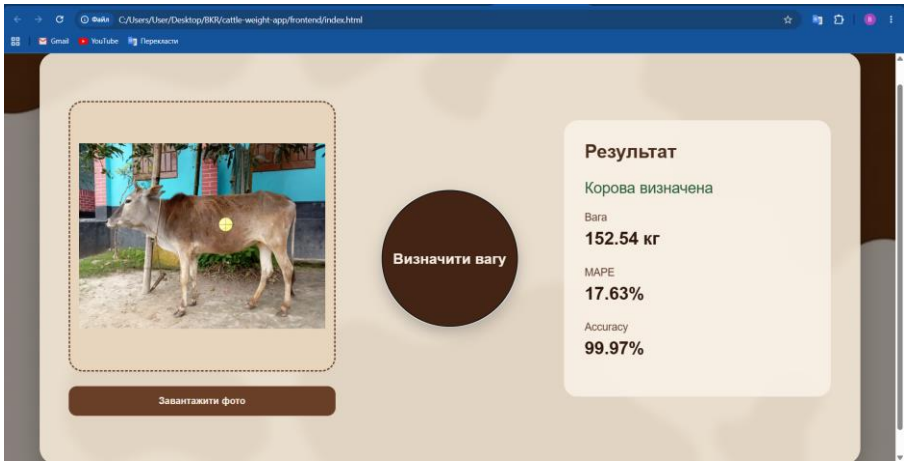


Рисунок 2. Результат роботи системи

Висновки. У результаті виконаної роботи розроблено інформаційну систему для автоматизованого визначення фізичних характеристик тварин із використанням технологій комп'ютерного зору та глибокого навчання. Практичну реалізацію системи виконано на прикладі задачі прогнозування ваги великої рогатої худоби за цифровими зображеннями.
Запропонований підхід дозволяє автоматизувати процес оцінювання фізичних параметрів тварин без використання спеціалізованого вимірювального обладнання, що спрощує моніторинг стану тварин у фермерських господарствах. Практичне значення роботи полягає у можливості подальшого масштабування системи для визначення інших характеристик тварин та інтеграції у системи Smart Farming.

Список літератури

1. Bezsonov O., Lebediev O., Lebediev V., Megel Y., Prochukhan D., Rudenko O. Breed recognition and estimation of live weight of cattle based on methods of machine learning and computer vision // Eastern-European Journal of Enterprise Technologies. 2021. Vol. 6, No. 9 (114). P. 64–74.

2. Güvenoğlu E. Determination of the Live Weight of Farm Animals with Deep Learning and Semantic Segmentation Techniques // Applied Sciences. 2023. Vol. 13, No. 12. Article 6944. DOI: 10.3390/app13126944.

3. Afridi H., Ullah M., Nordbø Ø., Hoff S. C., Furre S., Larsgard A. G., Cheikh F. A. Analyzing Data Modalities for Cattle Weight Estimation Using Deep Learning Models // Journal of Imaging. 2024. Vol. 10, No. 3. Article 72. DOI: 10.3390/jimaging10030072

Askerov V.V.
Postgraduate Student of Computer Engineering and Information Systems Department, Khmelnytskyi National University

RISK-AWARE ARCHITECTURE FOR ADAPTIVE MANAGEMENT OF SECURE TRANSACTIONS IN PERMISSIONED BLOCKCHAIN SYSTEMS

Introduction. Blockchain systems are widely used as a technological basis for transaction processing in financial services, inter-organizational registries, supply-chain infrastructures, and distributed digital platforms. Their key advantages are cryptographic integrity, immutability of records, distributed verification, and resistance to unauthorized modification of already confirmed data. However, the cryptographic correctness of a transaction and its inclusion in a valid block do not automatically mean that the transaction is safe from the application security perspective. Fraudulent schemes, money laundering, anomalous chains of addresses, transaction splitting, and coordinated malicious activity may still be present even when the transaction technically satisfies the formal validation rules of the blockchain network [1].

Modern research confirms the relevance of machine learning and graph-based models for detecting suspicious blockchain transactions. In particular, recurrent graph convolutional models are used for sequential prediction of illicit cryptocurrency transactions, where both graph structure and temporal dynamics are important for risk identification [2]. At the same time, transaction fraud detection is complicated by class imbalance, because dangerous transactions usually represent a minority class and require special attention to precision, recall, and interpretability of the obtained results [3].

This creates a methodological and engineering gap. A blockchain system requires not only risk detection but also a controlled mechanism for using the detected risk class in the transaction execution process. Such a mechanism must preserve the determinism of the blockchain network, avoid modification of the basic consensus algorithm, and provide a verifiable explanation of why a particular transaction was admitted, delayed, additionally verified, or blocked. Therefore, the development of a risk-aware architecture for adaptive management of secure transactions in permissioned blockchain systems is an relevant scientific and applied task.

Problem Statement. The object of the research is the process of secure transaction management in permissioned blockchain systems. The subject of the research is an architectural model that combines standard on-chain transaction processing with a trusted off-chain risk assessment component for adaptive control of transaction admission and verification.

The aim of the research is to improve the architecture of a hardware-software blockchain system for secure transactions by introducing a security component that determines the transaction risk class and adaptively corrects the transaction execution protocol without modifying the basic consensus mechanism.

To achieve this aim, it is necessary to solve the following tasks: to define the functional composition of the risk-aware blockchain architecture; to separate the on-chain transaction processing layer from the trusted off-chain risk assessment layer; to formalize the role of the neural network risk assessment component in the transaction management process; to define the adaptive control logic for transactions of different risk classes; and to ensure that the off-chain decision can be verified, explained, and used for further audit.

Main Material. The proposed architecture is based on the integration of two logically separated but functionally connected parts: the on-chain blockchain network and the trusted off-chain risk assessment circuit. The on-chain part includes the standard components of a permissioned blockchain system: transaction formation, transaction pool or mempool, pre-consensus validation, consensus mechanism, block formation, and distributed ledger [4]. These components remain unchanged and operate according to the basic protocol of the blockchain platform.

The off-chain part is introduced as an additional security circuit that performs intelligent risk assessment before the transaction is finally processed by the blockchain network. This circuit includes transaction feature extraction, neural network risk scoring, uncertainty estimation, local explanation of

the decision, policy selection, and formation of a signed decision record. The main architectural principle is that computationally complex and potentially non-deterministic risk analysis is performed outside the blockchain, while the blockchain itself is used for deterministic execution, record immutability, and cryptographic verification of the decision metadata.

Let the transaction be denoted as *tx*, and the corresponding feature vector as $x(tx)$. The feature vector may include structural, temporal, and contextual characteristics of the transaction. Structural features describe the position of the transaction in the transaction graph, the relations between addresses, and the properties of incoming and outgoing links. Temporal features reflect the time of transaction execution, intervals between related operations, and dynamic changes in transaction behavior. Contextual features characterize additional metadata, previous activity patterns, and possible relations with known suspicious transaction chains.

The off-chain neural network component calculates a risk score:

$$r = f_\theta\big(x(tx)\big),$$

where f_θ is the trained neural network model with parameters θ, and r is the estimated level of transaction risk. In addition to the risk score, the architecture provides the estimation of prediction uncertainty:

$$u = g\big(r, x(tx)\big),$$

where u characterizes the stability of the model decision. This is important because not all transactions should be processed only by a binary rule "safe / unsafe". Some transactions may belong to an intermediate zone where the model confidence is insufficient for direct automatic admission or direct blocking.

The architecture uses three transaction risk classes: WHITE, GRAY, and BLACK. The WHITE class corresponds to transactions with low risk and stable prediction. The GRAY class corresponds to transactions with intermediate risk or increased uncertainty. The BLACK class corresponds to transactions with high risk and sufficiently stable model decision. The assignment of a transaction to a particular class may be described by threshold rules:

$$WHITE: r \le t_{\{white\}}, u \le u_{\{max\}}$$
$$BLACK: r \ge t_{\{black\}}, u \le u_{\{max\}},$$
$$GRAY: otherwise.$$

Here $t_{\{white\}}$, $t_{\{black\}}$, and $u_{\{max\}}$ are threshold parameters selected on the validation set according to the required trade-off between transaction processing speed, false positive rate, and security level.

For each risk class, a separate transaction management policy is applied. Transactions classified as WHITE are processed in the standard mode. They receive normal priority, pass ordinary pre-consensus validation, and are routed to the mempool for further processing by the consensus mechanism. This mode minimizes delays and preserves the normal throughput of the blockchain network.

Transactions classified as GRAY are processed in the enhanced verification mode. Such transactions may receive lower priority, be temporarily delayed, or be sent for additional policy checks before inclusion in the mempool. This allows the system to reduce the probability of incorrect automatic decisions for borderline or uncertain cases. The GRAY zone is especially important because it prevents the architecture from behaving as an overly rigid binary filter. Instead of immediate blocking, the system introduces a controlled verification stage.

Transactions classified as BLACK are processed in the restrictive mode. They may be blocked before mempool admission, placed in quarantine, or marked for incident investigation. This makes it possible to prevent potentially dangerous transactions from entering the normal execution path of the blockchain network. At the same time, the consensus mechanism itself is not modified: the adaptive correction is implemented at the level of admission, routing, and verification policies.

An important feature of the proposed architecture is the preservation of the basic blockchain consensus. The architecture does not require changing the algorithm by which network nodes reach agreement on the state of the ledger. Instead, it introduces a pre-consensus security layer that changes only the mode of transaction handling before the transaction becomes part of the ordinary consensus process. This makes the proposed solution compatible with permissioned blockchain networks, where

governance rules, trusted participants, and access policies can support the use of an external risk assessment circuit.

To ensure trust in off-chain decisions, the architecture includes a mechanism for forming a signed decision record. Such a record may include the transaction identifier, the determined risk class, the hash of the feature vector, the version identifier of the neural network model, the version identifier of the applied policy, the hash of the local explanation, the timestamp, and the digital signature of the trusted risk assessment component. The full off-chain analysis does not need to be stored directly in the blockchain. Instead, cryptographically linked metadata are bound to the transaction or to the decision log. This provides the possibility of verifying that the decision was made using a specific model version, a specific feature set, and a specific security policy.

The architectural novelty of the proposed approach is that the neural network component is not used only as an isolated detector. It becomes a functional element of the transaction management architecture. The risk class produced by the model directly determines the execution policy: standard validation, enhanced verification, delayed admission, blocking, or quarantine. Thus, the architecture transforms machine learning output into an adaptive control signal for secure transaction processing.

Another important aspect is explainability. Since transaction blocking or additional verification may have operational consequences, the system must provide not only the final class but also a traceable explanation of the decision. Local explanation methods can identify the most influential transaction features and support subsequent audit. In this way, the architecture combines adaptive security, operational control, and interpretability.

Experimental verification of the corresponding neural network component on the Elliptic Bitcoin Dataset showed the practical applicability of the proposed risk-aware approach. The use of temporal splitting for training, validation, and testing allows the evaluation to be closer to real operating conditions, where transaction patterns may change over time. The obtained values of ROC-AUC and AUPRC confirm that the neural network component is capable of separating risky transactions under class imbalance. However, the main result of the proposed architecture is not limited to classification quality. The principal contribution is the integration of risk assessment into the transaction execution logic of a permissioned blockchain system without changing its consensus mechanism.

Conclusions. The paper proposes a risk-aware architecture for adaptive management of secure transactions in permissioned blockchain systems. The architecture improves the traditional blockchain transaction processing model by adding a trusted off-chain security component for determining the transaction risk class and adaptively correcting the transaction execution protocol.

The scientific novelty of the proposed approach consists in the architectural integration of neural network risk assessment with transaction admission and verification policies. In contrast to approaches that only classify transactions as suspicious or legitimate, the proposed architecture introduces adaptive transaction handling based on the WHITE, GRAY, and BLACK risk classes. This makes it possible to apply different modes of transaction processing: standard validation, enhanced verification, delayed admission, blocking, or quarantine.

The practical significance of the proposed architecture is that it increases the security level of transaction processing while preserving the standard mechanisms of the blockchain network. The basic consensus algorithm is not modified, and adaptive control is implemented at the level of pre-consensus admission, routing, and policy-based verification. The use of signed decision records and cryptographically linked metadata creates the basis for auditability, reproducibility, and verification of off-chain risk assessment results.

References

1. Elmougy Y., Liu L. Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics. *Proceedings of the ACM Web Conference 2023*. 2023. DOI: 10.1145/3580305.3599803.

2. Alarab I., Prakoonwit S. Robust recurrent graph convolutional network approach based sequential prediction of illicit transactions in cryptocurrencies. *Multimedia Tools and Applications*. 2024. Vol. 83. P. 58449–58464. DOI: 10.1007/s11042-023-17323-4.

3. Yin P., Jiang W., Ma Z., Zhang L. Cryptocurrency Transaction Fraud Detection Based on Imbalanced Classification With Interpretable Analysis. *International Journal of Intelligent Information Technologies*. 2024. Vol. 20, Issue 1. DOI: 10.4018/IJIIT.357696.

4. Askerov V., Tomchyshen B., Bouhissi H. (2025). Use of smart contracts on the ton blockchain for innovative educational solutions development. *Computer Systems and Information Technologies*, (1), 147–155. DOI: 10.31891/csit-2025-1-17

Віт Р.В.
аспірант четвертого курсу, молодший науковий співробітник Хмельницький національний університет,
Наук. керівник Собко О.В. доктор філософії з комп'ютерних наук, старший викладач кафедри КН Хмельницький національний університет

ФОРМУВАННЯ РЕПРЕЗЕНТАТИВНОГО НАБОРУ МАКРОЗОБРАЖЕНЬ ТКАНИН ДЛЯ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ЦИРКУЛЯРНІЙ ЕКОНОМІЦІ

Вступ. Текстильна промисловість є однією з галузей, у яких проблема відходів безпосередньо пов'язана з неефективністю сортування, складністю ідентифікації сировинного складу матеріалів та обмеженими можливостями їх повторного використання. У контексті циркулярної економіки важливим є не лише зменшення кількості текстильних відходів, а й підвищення точності їх розподілу за типом матеріалу, оскільки від правильності класифікації залежить подальший вибір технології переробки або повторного використання.

Сучасні дослідження підтверджують активне застосування методів штучного інтелекту, машинного навчання, глибинного навчання та комп'ютерного зору для автоматизації сортування текстильних відходів [1]. Зокрема, системи машинного зору з механізмами уваги використовуються для класифікації вживаного одягу, що демонструє потенціал автоматизованих підходів для зменшення залежності від ручної праці [2]. Окремий напрям становлять рішення на основі зображень тканин, отриманих за допомогою недорогого оптичного обладнання, що дозволяє розглядати комп'ютерний зір як альтернативу дорогим спектральним методам у задачах попередньої ідентифікації матеріалів [3].

Водночас ефективність моделей штучного інтелекту для класифікації текстильних матеріалів значною мірою залежить від якості, обсягу та репрезентативності навчальних даних. Наявність окремих моделей класифікації без відкритого, структурованого і відтворюваного набору зображень не забезпечує достатньої наукової перевірюваності результатів. Тому формування репрезентативного набору макрозображень тканин із метаданими є самостійною науково-прикладною задачею, що створює основу для подальшого навчання, порівняння та валідації нейромережевих моделей у задачах автоматизованого сортування текстильних відходів.

Постановка задачі. Об'єктом дослідження є процес підготовки візуальних даних для нейромережевої класифікації текстильних матеріалів у системах циркулярної економіки. Предметом дослідження є підхід до формування репрезентативного набору макрозображень тканин, придатного для навчання, тестування та порівняльного оцінювання моделей штучного інтелекту.

Метою роботи є обґрунтування та розроблення підходу до формування відкритого репрезентативного набору макрозображень тканин із метаданими для подальшого застосування в задачах нейромережевої класифікації текстильних відходів.

Для досягнення поставленої мети необхідно розв'язати такі задачі: визначити вимоги до структури набору макрозображень тканин; обґрунтувати класову структуру даних для задачі сортування текстильних відходів; сформувати протокол отримання зображень у контрольованих умовах; визначити склад метаданих, необхідних для відтворюваності дослідження; запропонувати критерії репрезентативності та контролю якості зображень;

описати придатність сформованого набору для подальшого навчання моделей штучного інтелекту.

Основний матеріал. Формування набору макрозображень тканин для задач штучного інтелекту доцільно розглядати не як допоміжний етап, а як базовий компонент усієї методології автоматизованої класифікації текстильних відходів. Причина полягає в тому, що нейромережева модель навчається не на абстрактних властивостях матеріалу, а на конкретних візуальних представленнях тканини, які залежать від умов зйомки, масштабу, освітлення, орієнтації волокон, кольору, фактури, зношеності та деформацій зразка.

У межах запропонованого підходу макрозображення тканини розглядається як цифрове представлення поверхневої та структурної інформації про матеріал, що може бути використане моделлю комп'ютерного зору для віднесення зразка до певної категорії. Для задач циркулярної економіки базовою класовою структурою доцільно визначити три категорії текстильних матеріалів: натуральні, синтетичні та змішані [4]. Така структура відповідає практичній логіці сортування, оскільки різні групи матеріалів потребують різних сценаріїв повторного використання, механічної переробки, хімічної переробки або відбракування.

Ключовою вимогою до набору даних є репрезентативність. У цій роботі репрезентативність розглядається як здатність набору макрозображень відображати реальну варіативність текстильних матеріалів у межах кожного класу. Для цього набір має включати зразки з різними типами переплетення, щільністю структури, кольором, фактурою, ступенем зношеності, напрямом волокон і станом поверхні. Особливо важливо враховувати не лише "ідеальні" зразки тканин, а й складні випадки, характерні для текстильних відходів: складки, потертості, деформації, часткові забруднення, неоднорідність поверхні та візуальну подібність між різними класами.

Протокол отримання зображень має забезпечувати сталість умов фіксації та можливість повторення експерименту. До основних параметрів такого протоколу належать тип оптичного пристрою, масштаб або збільшення, роздільна здатність зображення, умови освітлення, відстань до зразка, фон, орієнтація тканини, сторона зразка, а також правила вибору області аналізу. Важливо, щоб зміни у зображеннях відображали властивості тканини, а не випадкові відмінності умов зйомки. Саме тому протокол повинен містити не лише технічні параметри зйомки, а й правила контролю якості зображень.

Окремим елементом підходу є формування структури метаданих. Зображення без супровідного опису має обмежену наукову цінність, оскільки втрачається інформація про походження зразка, умови фіксації та належність до класу. Доцільно передбачити такі групи метаданих: унікальний ідентифікатор зразка; клас матеріалу; заявлений або встановлений сировинний склад; тип тканини або її умовний опис; колір; сторона зйомки; параметри фіксації; дата створення зображення; ознаки попередньої обробки; показники якості; джерело зразка; версія набору даних. Така структура дозволяє використовувати набір не лише для навчання моделей, а й для подальшого аналізу помилок класифікації.

Контроль якості зображень є необхідним для зменшення впливу шумових факторів на навчання моделей. До ознак, що потребують перевірки, належать розмиття, недостатня різкість, нерівномірне освітлення, пересвічені ділянки, затемнення, наявність сторонніх об'єктів, неповне потрапляння зразка в область аналізу та дублювання майже однакових кадрів. Наявність таких артефактів не завжди означає автоматичне вилучення зображення, але потребує маркування у метаданих. Це дозволяє надалі формувати як "чисті" підвибірки для базового навчання, так і складні підвибірки для перевірки стійкості моделей.

Важливою методичною вимогою є коректний поділ набору на навчальну, валідаційну та тестову підвибірки. Такий поділ має виконуватися не лише на рівні окремих зображень, а передусім на рівні зразків тканин. Якщо зображення одного й того самого зразка потраплять одночасно до навчальної і тестової вибірок, це може створити ефект витоку даних і штучно завищити якість класифікації. Тому для об'єктивного оцінювання узагальнюючої здатності моделей необхідно забезпечити, щоб тестова підвибірка містила зразки, які не використовувалися під час навчання.

Сформований набір макрозображень може бути використаний для кількох типів задач штучного інтелекту. По-перше, він є основою для навчання згорткових нейронних мереж і трансформерних архітектур у задачі класифікації тканин за сировинною групою. По-друге, він може використовуватися для оцінювання стійкості моделей до змін кольору, фактури, орієнтації та деформацій. По-третє, на його основі можна будувати модулі валідації результатів, які визначають зразки з низькою впевненістю прогнозу та спрямовують їх на повторну перевірку або ручний контроль.

Аналіз сучасних досліджень показує, що автоматизоване сортування текстильних відходів поступово переходить від ізольованих моделей класифікації до комплексних систем, які поєднують комп'ютерний зір, штучний інтелект, контроль якості та інтеграцію в технологічні процеси [5]. Водночас відкритість даних, наявність метаданих і стандартизований протокол формування корпусу залишаються критичними умовами відтворюваності результатів. Попередні дослідження з класифікації натуральних і синтетичних тканин за зображеннями у видимому спектрі підтверджують перспективність такого напряму, але також вказують на потребу розширення корпусів даних, урахування складних випадків і розвитку багатокласової постановки задачі [6].

Запропонований підхід до формування набору макрозображень тканин передбачає, що датасет має бути відкритим, версіонованим і придатним до незалежної перевірки. Для цього доцільним є оприлюднення набору з DOI, ліцензією відкритого доступу, описом протоколу збору, структурою метаданих і документацією щодо класів. Такий формат забезпечує відповідність принципам FAIR, оскільки дані стають доступними для пошуку, повторного використання, порівняння результатів і подальшого розширення.

Висновки. У роботі обґрунтовано підхід до формування репрезентативного набору макрозображень тканин для застосування штучного інтелекту в задачах циркулярної економіки. Показано, що якість і структура даних є визначальними передумовами для подальшої побудови надійних нейромережевих моделей класифікації текстильних відходів.

Наукова новизна роботи полягає у формалізації вимог до відкритого набору макрозображень тканин як основи для нейромережевої класифікації матеріалів. Запропоновано враховувати не лише класову належність зразків, а й умови зйомки, структуру метаданих, критерії репрезентативності, контроль якості зображень і правила формування навчальної, валідаційної та тестової підвибірок без витоку даних.

Практичне значення роботи полягає у створенні методичної основи для формування відкритого датасету текстильних матеріалів, придатного для навчання, порівняння та валідації моделей комп'ютерного зору. Такий набір може бути використаний у подальших дослідженнях нейромережевої класифікації натуральних, синтетичних і змішаних тканин, а також у розробленні програмних модулів для автоматизованого сортування текстильних відходів.

Список літератури

1. Faghih E., Wallace M., Lin S.-H. A Systematic Literature Review – AI-Enabled Textile Waste Sorting. *Sustainability*. 2025. Vol. 17, Issue 10. Article 4264. DOI: 10.3390/su17104264.

2. Tian R., Lv Z., Fan Y., Wang T., Sun M., Xu Z. Qualitative classification of waste garments for textile recycling based on machine vision and attention mechanisms. *Waste Management*. 2024. Vol. 183. P. 74–86. DOI: 10.1016/j.wasman.2024.04.040.

3. Gupta D., Dubey A. Towards Improved Sustainability in the Textile Lifecycle with Deep Learning. *Proceedings of the 7th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*. 2024. DOI: 10.1145/3674829.3675077.

4. Derzhak V.V., Mazurets O.V. Approach to image preprocessing for household waste classification in circular economy. *Resource-Saving Technologies of Apparel, Textile & Food Industry.* Proceedings of International Scientific and Practical Conference. November 20, 2025. Khmelnytskyi, Ukraine. Pp. 319-323.

5. Weimer J., Evers N., Dumitrescu R. AI-Based System for Enhanced Textile Recycling and Reuse. In: *Advances in Information and Communication*. Springer, 2025. DOI: 10.1007/978-3-031-93891-7_69.

6. Mazurets O., Zalutska O., Molchanova M., Sobko O., Bukhantsova L., Zakharkevich O. Deep Learning-Driven Fabric Classification: Distinguishing Natural and Synthetic Materials. *CEUR Workshop Proceedings*, 2025, vol. 4164, pp. 39-51.

Тимофієв І.А.
аспірант першого курсу кафедри комп'ютерних наук Хмельницький національний університет
Наук керівник канд. техн. наук, доцент кафедри комп'ютерних наук Хмельницький національний університет Мазурець О.В.

ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ВИЯВЛЕННЯ СОЦІАЛЬНО-ДЕСТРУКТИВНИХ І ДЕПРЕСИВНИХ НАРАТИВІВ У ТЕКСТОВОМУ КОНТЕНТІ

Вступ. У сучасному соціальному, освітньому та цифровому середовищі значно зростає вплив факторів психологічного навантаження, зокрема стресу, емоційного виснаження, тривожності та інформаційного перенасичення [1]. Тривалий вплив таких чинників може спричиняти формування депресивних станів, а також поширення соціально-деструктивних наративів у цифровому комунікаційному просторі. Особливо актуальною ця проблема є для студентської молоді та активних користувачів онлайн-платформ, які перебувають в умовах інтенсивного навчального процесу, високих академічних вимог і недостатнього часу для психологічного відновлення та саморегуляції [2]. Своєчасне виявлення ознак депресивних проявів і деструктивної поведінки дає змогу оперативно забезпечити психологічну підтримку та знизити ризик розвитку більш складних психічних порушень [3].

Активне поширення цифрових комунікаційних платформ, соціальних мереж і текстових сервісів створює значні обсяги текстового контенту, який може містити маркери емоційного стану користувачів [4]. У таких умовах методи обробки природної мови (NLP) стають ефективним інструментом для автоматизованого аналізу психологічного стану людини, виявлення депресивних тенденцій та ідентифікації соціально-деструктивних наративів у текстових повідомленнях. Використання інтелектуальних інформаційних систем для аналізу текстового контенту забезпечує можливість своєчасного реагування на потенційно небезпечні психологічні прояви та сприяє підвищенню ефективності профілактичних заходів у сфері психічного здоров'я.

Постановка задачі. Метою роботи є розробка інтелектуальної інформаційної системи для автоматизованого виявлення соціально-деструктивних і депресивних наративів у текстовому контенті із застосуванням сучасних методів машинного навчання та технологій обробки природної мови.

Основний матеріал. Розробка інтелектуальної інформаційної системи для автоматизованого виявлення соціально-деструктивних і депресивних наративів базується на використанні відповідного розробленого методу [5]. Метод виявлення депресивних і соціально-деструктивних проявів базується на аналізі текстових даних за допомогою нейромережної моделі, здатної перетворювати вхідний текст у числове представлення, яке характеризує ймовірність наявності відповідних психологічних та поведінкових ознак. Основою методу є використання дуальної архітектури трансформерного типу, що поєднує можливості моделей BERT і GPT-2 для комплексного аналізу текстового контенту [6]. Такий підхід дозволяє одночасно враховувати синтаксичні, контекстні та семантичні особливості повідомлень користувача.

Вхідними даними системи є текстові повідомлення користувачів, які проходять попередню обробку та подальший аналіз за допомогою навченої нейромережної моделі. На першому етапі здійснюється токенізація тексту із використанням спеціалізованих токенізаторів, що відповідають архітектурам BERT та GPT-2 і були використані під час навчання моделей [7]. Це забезпечує коректне представлення тексту у форматі, придатному для подальшої обробки нейронними мережами.

Наступним етапом є паралельний аналіз токенізованих даних двома трансформерними моделями. Модель BERT виконує аналіз синтаксичних залежностей і контекстних зв'язків між елементами тексту, тоді як GPT-2 зосереджується на виявленні семантичних характеристик, емоційних маркерів і прихованих змістових закономірностей. Поєднання результатів роботи

обох моделей забезпечує підвищення точності визначення депресивних та соціально-деструктивних наративів.

На третьому етапі результати обробки інтегруються за допомогою спеціального шару об'єднання ознак, який формує підсумкове числове значення. Отримана оцінка відображає рівень ймовірності наявності депресивних проявів або соціально-деструктивного змісту в аналізованому тексті. Вихідними даними системи є інтегральний показник, що характеризує ступінь вираженості відповідних психологічних і поведінкових ознак у текстовому контенті користувача.

Після завершення етапу аналізу запускається вебсервер Flask у режимі налагодження, що забезпечує автоматичне перезавантаження застосунку під час внесення змін до програмного коду та надає детальну інформацію щодо можливих помилок виконання. Розроблений вебзастосунок реалізує базові функції інтелектуальної інформаційної системи, зокрема аналіз текстового контенту засобами моделей глибокого навчання, обробку результатів класифікації та управління текстовою базою даних через зручний вебінтерфейс, наведений на рисунку 1.

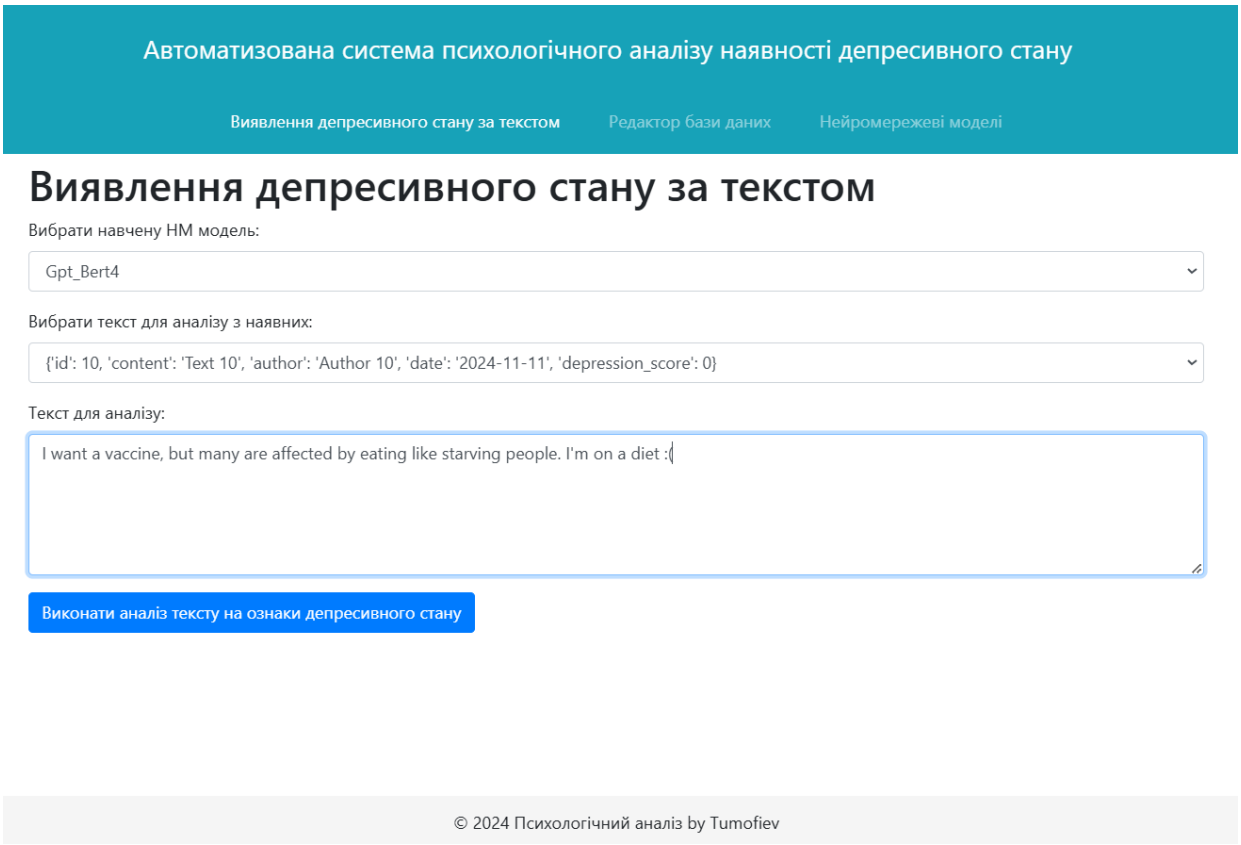


Рисунок 1 – Вигляд інтелектуальної інформаційної системи для автоматизованого виявлення соціально-деструктивних і депресивних наративів у текстовому контенті

Для дослідження ефективності автоматизованого виявлення соціально-деструктивних і депресивних наративів у текстовому контенті було проведено експериментальне порівняння архітектур-трансформерів GPT-2, BERT та розробленої нейромережевої моделі дуальної архітектури. Експеримент було проведено на 20 текстах, сформованих з використанням мовної моделі GPT 3.5, де 10 текстів з проявами депресії та 10 без проявів депресії.

Також було досліджено вплив параметрів на здатність нейромережевих моделей до навчання. Результати проведення дослідження для 4-х альтернативних моделей дуальної архітектури наведені на рисунку 2.

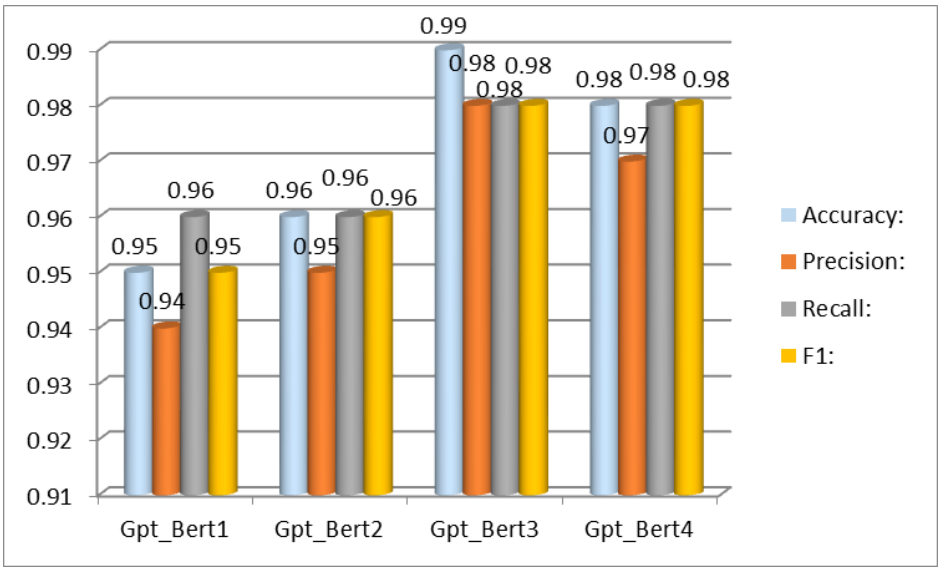


Рисунок 2 – Результати навчання нейромережевих моделей за метриками

Висновки. Таким чином, у роботі розглянуто особливості проєктування та реалізації прикладних компонентів інтелектуальної інформаційної системи автоматизованого виявлення соціально-деструктивних і депресивних наративів у текстовому контенті із використанням сучасних методів NLP та моделей глибокого навчання. Запропонований підхід дозволяє підвищити ефективність моніторингу психологічного стану користувачів цифрових платформ і створює передумови для подальшого розвитку систем підтримки психічного здоров'я.

Список літератури:

1. Jain P., Ram Srinivas K., Vichare A. Depression and Suicide Analysis Using Machine Learning and NLP. *Journal of Physics: Conference Series*. 2022. Vol. 2161, no. 1. P. 012034. URL: https://doi.org/10.1088/1742-6596/2161/1/012034

2. Mali A., Sedamkar R. R. Prediction of Depression Using Machine Learning and NLP Approach. *Intelligent Computing and Networking*. Singapore, 2022. P. 172–181. URL: https://doi.org/10.1007/978-981-16-4863-2_15

3. Lorenzoni G., Tavares C., Nascimento N., Alencar P., Cowan D.Assessing ML Classification Algorithms and NLP Techniques for Depression Detection: An Experimental Case Study. URL: https://arxiv.org/abs/2404.04284

4. Applying natural language processing to patient messages to identify depression concerns in cancer patients / M. M. van Buchem et al. *Journal of the American Medical Informatics Association*. 2024. URL: https://doi.org/10.1093/jamia/ocae188

5. O. Mazurets, R. Vit, M. Molchanova, I. Tymofiiev, O. Sobko, Context-enriched approach to students depression monitoring in education using BERT-GPT hybrid model, *CEUR Workshop Proceedings*. Vol. 4096 (2025) 167-176.

6. Тимофієв І.А., Мазурець О.В. Нейромережеве виявлення депресивних патернів соціально-деструктивних наративів у текстах із візуальним поясненням результатів. *Інформаційні технології і автоматизація*. Матеріали XVIII міжнародної науково-практичної конференції. 30-31 жовтня 2025 р. Одеса, ОНТУ. 2025. С.1066-1069.

7. Krak, I., Sobko, O., Mazurets, O., Tymofiiev, I., Molchanova, M., Barmak, O. Method for Detecting and Classifying Cyberbullying in Text Content Using Neural Networks. *Lecture Notes in Networks and Systems*. Springer, Cham. 2025. Vol. 1473. pp. 486–498.

Шурипа М.О.
студент першого курсу магістратури кафедри КН Хмельницький національний університет,
Наук. керівник к.т.н, доцент кафедри КН Хмельницький національний університет
Мазурець О.В.

МЕТОД ВАЛІДАЦІЇ АНТРОПОМЕТРИЧНОЇ ПОЗИ ЛЮДИНИ ДЛЯ ФОТОГРАММЕТРИЧНОГО ЗНЯТТЯ МІРОК НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ

Вступ. Розвиток технологій комп'ютерного зору та штучного інтелекту створює передумови для автоматизації процесів, які традиційно потребували ручного вимірювання та участі фахівця. Одним із таких напрямів є фотограмметричне зняття антропометричних мірок людини за фотозображеннями. Такий підхід є перспективним для задач добору розміру одягу, персоналізованого проєктування виробів, дистанційного антропометричного аналізу та цифрових сервісів у сфері легкої промисловості.

Однак точність подальшого визначення антропометричних параметрів значною мірою залежить не лише від алгоритму обчислення мірок, а й від якості вхідного фотоматеріалу. Якщо людина розташована під неправильним кутом, неповністю потрапляє в кадр, нахиляє корпус, згинає руки або ноги, повертає голову, стоїть у нестабільній позі чи перебуває в умовах недостатнього освітлення, подальший вимірювальний аналіз може бути некоректним. У такому випадку похибка виникає ще до початку основної обробки зображення.

Сучасні методи оцінювання пози людини, зокрема MediaPipe Pose та похідні рішення на основі BlazePose, дають змогу виявляти ключові точки тіла у режимі реального часу та використовувати їх для аналізу положення людини на зображенні [1; 2]. У роботах останніх років такі інструменти застосовуються для задач оцінювання пози, контролю рухів, аналізу суглобових кутів і побудови прикладних систем комп'ютерного зору [3, 4]. Водночас для задач фотограмметричного зняття мірок важливим є не лише виявлення людини в кадрі, а й перевірка відповідності її положення спеціалізованим вимогам антропометричної зйомки.

Тому актуальною є задача розроблення методу автоматизованої валідації антропометричної пози людини, який дозволяє ще на етапі отримання фото визначити, чи є зображення придатним для подальшого фотограмметричного зняття мірок. Такий метод має поєднувати нейромережеве визначення ключових точок тіла з формалізованими геометричними правилами контролю анфасної та профільної пози.

Постановка задачі. Об'єктом дослідження є процес підготовки фотозображень людини для подальшого фотограмметричного зняття антропометричних мірок. Предметом дослідження є метод автоматизованої валідації антропометричної пози людини на основі комп'ютерного зору.

Метою роботи є розроблення методу валідації антропометричної пози людини, який забезпечує автоматизовану перевірку придатності фотозображення для подальшого фотограмметричного зняття мірок за анфасним і профільним сценаріями зйомки.

Для досягнення поставленої мети необхідно розв'язати такі задачі: визначити типові помилки фотозйомки, що унеможливлюють коректне зняття мірок; сформувати набір критеріїв перевірки анфасної та профільної пози; використати ключові точки скелетної моделі людини для оцінювання положення тіла; розробити правила контролю повноти потрапляння людини в кадр, положення голови, рук, ніг і корпусу; забезпечити формування статусу валідації та рекомендацій користувачу щодо виправлення пози.

Основний матеріал. Запропонований метод ґрунтується на поєднанні нейромережевого визначення пози людини та системи геометричних правил, орієнтованих на вимоги антропометричної фотозйомки. На відміну від підходів, у яких комп'ютерний зір використовується безпосередньо для розрахунку розмірів тіла, у цій роботі основна увага зосереджена на попередній валідації фотозображення. Тобто метод не виконує обчислення антропометричних мірок, а визначає, чи може отримане зображення бути використане для такого обчислення на наступному етапі.

Вхідними даними методу є фотозображення або відеокадр, на якому зображена людина. За допомогою модуля оцінювання пози визначаються ключові точки тіла: голова, очі, вуха, плечі, лікті, кисті, таз, коліна, щиколотки, п’яти та стопи. Для кожної точки враховуються її координати та показник достовірності виявлення. Це дозволяє оцінити не лише геометричне положення частин тіла, а й надійність отриманих даних.

Загальну логіку методу можна подати як послідовність таких етапів: отримання зображення; попередня обробка кадру; виявлення ключових точок тіла; перевірка якості зображення; визначення сценарію зйомки; застосування правил валідації пози; формування рішення про придатність або непридатність кадру; виведення рекомендацій користувачу. Така послідовність дозволяє розглядати валідацію пози як окремий контрольний контур перед подальшим вимірювальним аналізом.

Для анфасного зображення метод перевіряє, чи людина розташована обличчям до камери, чи повністю потрапляє в кадр, чи видимі основні опорні точки тіла, чи не нахилені плечі, чи не зігнуті ноги, чи не розміщені руки занадто близько до тулуба або надмірно відведені в сторони. Особливе значення має положення рук, оскільки притиснуті до тулуба руки можуть спотворювати силует і ускладнювати подальше визначення ширинних параметрів тіла. Тому для анфасної пози доцільно перевіряти кут відведення руки від вертикалі за координатами плеча та ліктя.

Для профільного зображення основним завданням є перевірка того, що людина дійсно розташована боком до камери. Для цього аналізується горизонтальне рознесення плечей і таза, фронтальність тулуба, асиметрія видимості лівої та правої сторін тіла, а також взаємне положення очей і вух. Додатково контролюється, щоб голова не була повернута до камери, корпус не мав надмірного нахилу, рука була опущена вздовж тулуба, ноги залишалися випрямленими, а стопи повністю потрапляли в кадр.

Формально рішення про придатність зображення можна подати як результат перевірки множини умов:

$$V=Q\wedge P\wedge F\wedge G\ ,$$

де V – результат валідації зображення; Q – виконання вимог до якості кадру; P – відповідність пози сценарію зйомки; F – повнота потрапляння тіла в кадр; G – виконання геометричних обмежень для ключових частин тіла. Якщо хоча б одна з критичних умов не виконується, зображення не допускається до подальшого фотограмметричного аналізу.

Для зменшення кількості випадкових помилок метод передбачає не лише одноразову перевірку кадру, а й контроль стабільності пози. У режимі роботи з камерою аналіз виконується циклічно. Якщо поза відповідає встановленим критеріям протягом заданого проміжку часу, запускається автоматичний зворотний відлік до фіксації фото. Безпосередньо в момент створення знімка перевірка повторюється. Це дає змогу уникнути ситуації, коли користувач проходить первинну перевірку, але змінює положення тіла під час відліку.

Важливою складовою методу є формування не лише бінарного рішення “придатне / непридатне”, а й пояснювального повідомлення для користувача. Якщо поза не відповідає вимогам, система має вказати причину відхилення: необхідність стати рівніше, повернутися обличчям до камери, розташуватися боком, не повертати голову, відвести руки від тіла, поставити ноги ширше, показати повний зріст або розмістити стопи в межах кадру. Такий підхід перетворює метод із пасивного фільтра зображень на керовану процедуру отримання коректного фотоматеріалу.

Окрему увагу приділено роботі із завантаженими статичними зображеннями [5]. У такому режимі метод виконує аналіз пози на готовому фото та допускає його до подальшого використання лише за умови проходження правил валідації. Для профільних зображень доцільною є також перевірка дзеркального відображення: якщо початковий варіант не проходить валідацію, система може проаналізувати горизонтально віддзеркалене зображення. Це важливо у випадках, коли орієнтація профільного знімка не відповідає очікуваному напрямку обробки. Приклад роботи прототипу, подубованого за наведеним методом показано на рисунку 1.

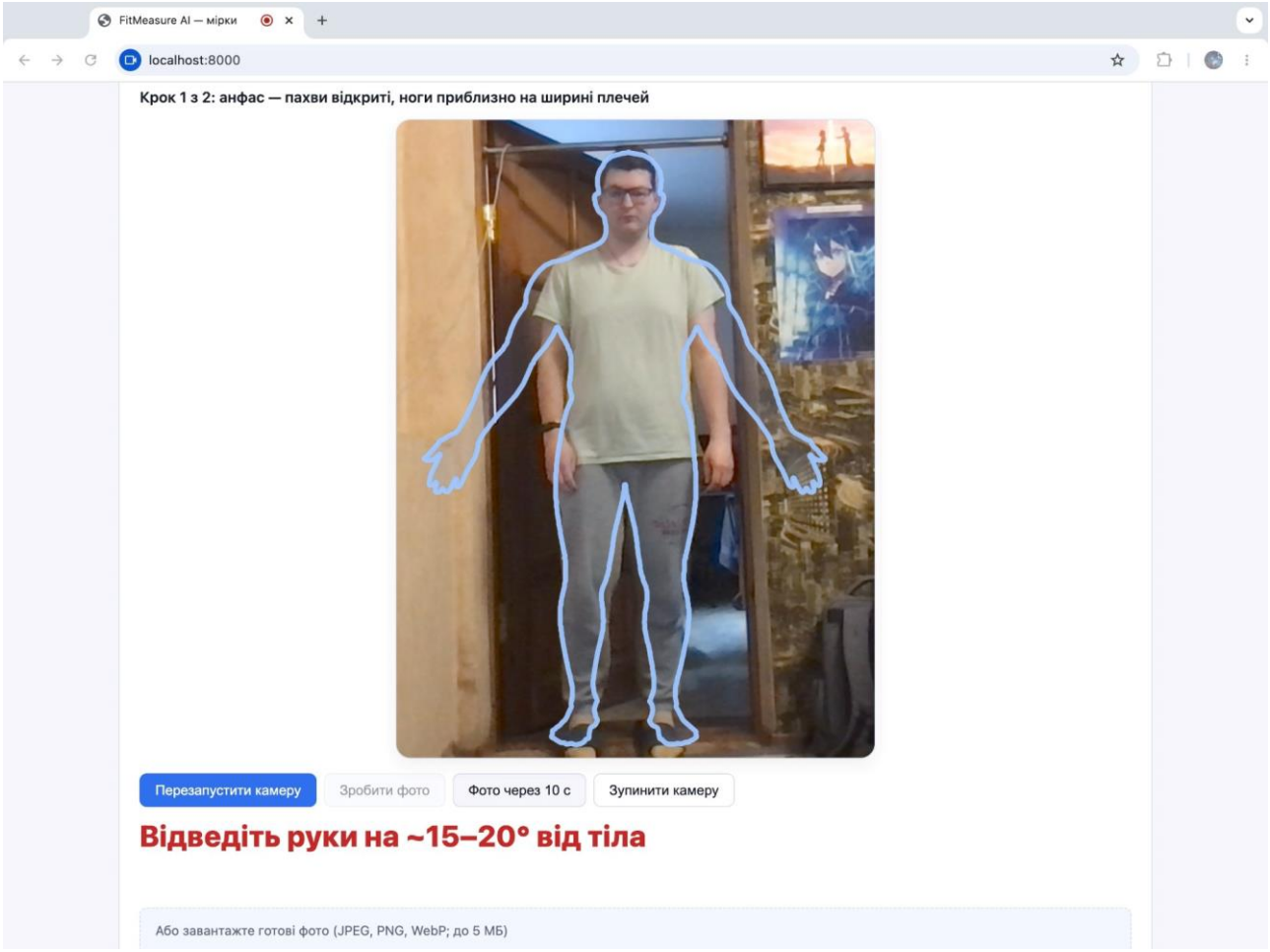


Рисунок 1 – Приклад непридатної для аналізу антропометричної пози

Після успішної перевірки зображення може бути приведене до уніфікованого формату. Нормалізація передбачає кадрування або доповнення до квадратної форми, масштабування до заданого максимального розміру та збереження в уніфікованому форматі. Це забезпечує стандартизованість вхідних даних для наступних етапів фотограмметричного аналізу та зменшує вплив випадкових відмінностей між зображеннями.

Наукова новизна запропонованого методу полягає в тому, що нейромережеве визначення пози людини використовується не як самодостатня задача розпізнавання скелета, а як основа для спеціалізованої валідації антропометричної пози. Метод поєднує координати ключових точок тіла, показники достовірності їх виявлення та набір геометричних правил, що враховують вимоги саме до анфасної та профільної фотозйомки для подальшого зняття мірок.

Практичне значення методу полягає у зменшенні кількості непридатних фотозображень ще до початку основного вимірювального аналізу. Це дозволяє підвищити стандартизованість вхідних даних, зменшити вплив помилок користувача, скоротити кількість повторних обчислень і створити передумови для точнішого фотограмметричного визначення антропометричних параметрів. Особливо важливо, що метод може бути використаний у веб- або мобільних сервісах, де користувач самостійно виконує фотозйомку без участі фахівця.

Висновки. У роботі запропоновано метод валідації антропометричної пози людини для фотограмметричного зняття мірок на основі комп'ютерного зору. Метод забезпечує попередню перевірку фотозображення перед його використанням у задачах антропометричного аналізу.

Науковий результат полягає у формалізації процедури перевірки придатності анфасних і профільних фотозображень за сукупністю ознак: якість кадру, повнота потрапляння тіла в кадр, положення голови, рук, ніг, корпусу, стабільність пози та відповідність сценарію зйомки. На відміну від загального виявлення людини на фото, запропонований метод орієнтований на спеціалізовані вимоги фотограмметричного зняття антропометричних мірок.

Практична цінність методу полягає в автоматизованому виявленні типових помилок зйомки та формуванні конкретних рекомендацій користувачу щодо їх усунення. Це дозволяє підвищити якість вхідного фотоматеріалу, зменшити кількість непридатних зображень і створити надійну основу для подальшого коректного визначення антропометричних параметрів людини.

Список літератури:

1. Bazarevsky V., Grishchenko I., Raveendran K., Zhu T., Zhang F., Grundmann M. BlazePose: On-device Real-time Body Pose Tracking. *arXiv preprint arXiv:2006.10204*. 2020.
2. Kim J.W., Choi H., Lee J. Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model. *Applied Sciences*. 2023. Vol. 13, Issue 4. Article 2700. DOI: 10.3390/app13042700.
3. Roggio F., Ravalli S., Maugeri G., Bianco A., Palma A., Di Rosa M., Musumeci G. A comprehensive analysis of the machine learning pose estimation models used in human movement analysis. *Heliyon*. 2024. Vol. 10, Issue 20. Article e38671. DOI: 10.1016/j.heliyon.2024.e38671.
4. Mazurets O., Zalutska O., Molchanova M., Sobko O., Bukhantsova L., Zakharkevich O. Deep Learning-Driven Fabric Classification: Distinguishing Natural and Synthetic Materials. CEUR Workshop Proceedings, 2025, vol. 4164, pp. 39-51.
5. Молчанова М.О., Савенко Б.О., Залуцька О.О., Мазурець О.В. Дослідження впливу аугментації зображень на точність нейромережевої класифікації текстильних відходів. Науковий журнал «Вісник Херсонського національного технічного університету». 2026. №2 (96). С. 288-294. https://doi.org/10.35546/kntu2078-4481.2026.2.35

Юрченко Д.Ю.
аспірант першого курсу кафедри комп'ютерних наук Хмельницький національний університет,
канд. техн. наук, доцент кафедри комп'ютерних наук Хмельницький національний університет Мазурець О.В.

ДОСЛІДЖЕННЯ МЕТОДУ ВІЗУАЛЬНОГО ПОЯСНЕННЯ РЕЗУЛЬТАТІВ НЕЙРОМЕРЕЖЕВОГО ВИЯВЛЕННЯ МАНІПУЛЯТИВНИХ ТЕХНІК У ТЕКСТОВОМУ КОНТЕНТІ

Вступ. У сучасному інформаційному просторі соціальні мережі, форуми, месенджери та інші цифрові комунікаційні платформи стали основними каналами поширення інформації та взаємодії між користувачами. Саме через текстовий контент сьогодні здійснюється значний вплив на громадську думку, формування соціальних настроїв і поведінкових моделей. Поряд із конструктивною комунікацією дедалі частіше спостерігається поширення маніпулятивних технік, спрямованих на емоційний вплив, викривлення сприйняття інформації, формування упереджень або стимулювання деструктивної поведінки. У зв'язку з цим особливої актуальності набуває розробка інтелектуальних методів автоматизованого виявлення маніпулятивного контенту та забезпечення прозорості роботи нейромережевих моделей, що використовуються для такого аналізу.

Однією з ключових проблем сучасних систем глибокого навчання є складність інтерпретації результатів їхньої роботи. Незважаючи на високу точність нейромережевих моделей при класифікації текстів, користувачі та аналітики часто не мають можливості зрозуміти, які саме фрагменти тексту або лінгвістичні ознаки вплинули на прийняття рішення моделлю. Це знижує рівень довіри до автоматизованих систем аналізу текстового контенту та ускладнює використання таких рішень у практичній діяльності. Саме тому методи візуального пояснення результатів нейромережевого аналізу стають важливим напрямом розвитку сучасних інтелектуальних інформаційних систем.

Аналіз текстового контенту та визначення маніпулятивних технік ґрунтуються на використанні методів комп'ютерної лінгвістики, машинного навчання та обробки природної

мови. Такі методи дозволяють автоматизовано виявляти емоційно забарвлену лексику, приховані смислові конструкції, психологічні тригери, маніпулятивні мовні шаблони та інші особливості тексту, які можуть впливати на сприйняття інформації аудиторією. Маніпулятивний вплив у текстовому контенті може реалізовуватися через використання емоційного тиску, нав'язування оцінок, перекручення фактів, поляризацію суджень або створення штучного інформаційного резонансу [1].

Під час взаємодії користувачів у соціально орієнтованих сервісах формується значний обсяг текстових повідомлень, які містять не лише семантичні, а й емоційні, психологічні та поведінкові характеристики. Це обумовлює необхідність розвитку інтелектуальних інформаційних технологій для автоматизованої обробки та аналізу текстового контенту, зокрема для виявлення маніпулятивних технік у повідомленнях користувачів [2]. Наявність таких систем дозволяє підвищити ефективність інформаційного моніторингу, забезпечити своєчасне реагування на деструктивний контент та покращити якість аналітичної підтримки під час роботи з великими обсягами даних.

Особливого значення набуває використання інструментів інтерпретаційної візуальної аналітики, які дозволяють представити результати роботи нейромережевих моделей у зрозумілій для користувача формі. Візуалізація ключових слів, фраз і ознак, що найбільше вплинули на класифікацію тексту, сприяє підвищенню прозорості системи та полегшує аналіз результатів для фахівців різних галузей, зокрема аналітиків, маркетологів, спеціалістів із кібербезпеки та дослідників соціальних комунікацій.

Постановка задачі. Метою роботи є прикладна реалізація та дослідження методу візуального пояснення результатів нейромережевого виявлення маніпулятивних технік у текстовому контенті із використанням сучасних методів глибокого навчання та інтерпретаційної аналітики.

Основний матеріал. Запропонований метод візуального пояснення результатів нейромережевого аналізу базується на використанні гібридної нейронної мережі, що поєднує архітектури CNN та BiLSTM, а також моделі локальної інтерпретації LIME. Загальна схема та основні етапи функціонування методу наведені на рисунку 1.

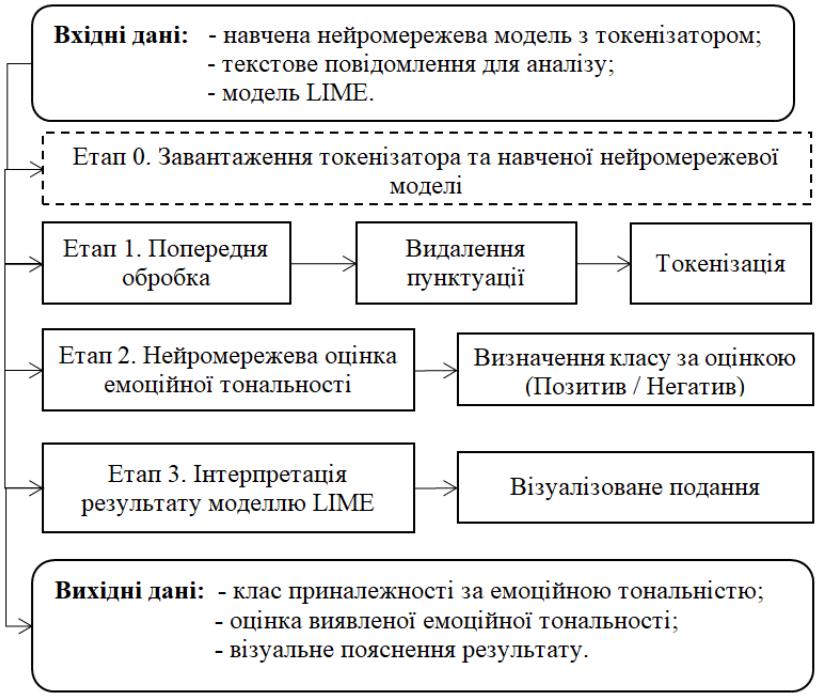


Рисунок 1 – Метод візуального пояснення результатів нейромережевого виявлення маніпулятивних технік у текстовому контенті

Поєднання архітектур CNN і BiLSTM забезпечує можливість комплексного аналізу текстового контенту. Згорткові нейронні мережі CNN ефективно виявляють локальні текстові

патерни, ключові слова, стійкі мовні конструкції та характерні фрагменти тексту, які можуть свідчити про використання маніпулятивних технік [3]. Водночас Bidirectional LSTM забезпечує аналіз довгострокових контекстних залежностей у двох напрямках, що дозволяє враховувати зміст повідомлення в цілому та підвищує точність класифікації.

Вхідними даними методу є попередньо навчена нейромережева модель із токенізатором, текстове повідомлення для аналізу та модель LIME, яка використовується для пояснення результатів класифікації. На першому етапі виконується завантаження токенізатора та навченої нейромережі, після чого здійснюється попередня обробка тексту. Вона включає нормалізацію тексту, приведення символів до нижнього регістру, видалення пунктуації, стоп-слів, службових символів та токенізацію відповідно до параметрів, використаних під час навчання моделі.

Наступним етапом є аналіз тексту за допомогою гібридної нейромережі CNN-BiLSTM. У процесі обробки модель визначає наявність ознак маніпулятивного впливу та формує числову оцінку ймовірності належності тексту до певного класу. Рішення про наявність або відсутність маніпулятивних технік приймається на основі порогового значення вихідного показника нейромережі. Третій етап передбачає інтерпретацію отриманого результату за допомогою методу LIME. Локальна інтерпретація дозволяє визначити, які саме слова, фрази або мовні конструкції найбільше вплинули на рішення моделі. Результати інтерпретації подаються у вигляді візуалізації, що демонструє вагомість окремих текстових елементів та ступінь їхнього впливу на класифікацію повідомлення.

Вихідними даними методу є клас належності тексту за ознакою наявності маніпулятивних технік, числова оцінка рівня маніпулятивного впливу, а також візуальне пояснення результату класифікації.

Програмна структура компонентів інформаційної системи візуального пояснення результатів нейромережевого аналізу складається з двох основних підсистем: вебзастосунку з графічним інтерфейсом користувача та окремого програмного модуля для навчання нейромережевих моделей без використання графічного інтерфейсу. Такий підхід забезпечує розподіл функцій між навчанням моделей та їх практичним використанням у процесі аналізу текстового контенту.

Вебзастосунок для інтерпретаційної візуальної аналітики реалізовано у вигляді набору взаємопов’язаних сторінок: «index.html», «nnanalize.html», «performance.html», «dataset.html», «detail.html» та «help.html». Схема переходів між сторінками наведена на рисунку 2.

Центральним елементом навігаційної структури є головна сторінка «index.html», яка виступає основною точкою взаємодії користувача із системою. Із неї здійснюється перехід до модулів аналізу текстового контенту, оцінки ефективності моделей, роботи з наборами даних та перегляду довідкової інформації. Сторінка «dataset.html» забезпечує доступ до детального перегляду окремих записів через сторінку «detail.html», що дозволяє виконувати більш глибокий аналіз вибраних прикладів. Розроблена структура навігації забезпечує інтуїтивно зрозумілий інтерфейс користувача, спрощує взаємодію із функціональними компонентами системи та підтримує логічну послідовність переходів між окремими модулями вебзастосунку. Після завершення навчання нейромережі виконується оцінювання моделі на тестових даних із використанням основних метрик класифікації, зокрема точності, повноти та F_1-міри. Для детального аналізу процесу навчання будуються графіки зміни функції втрат і точності моделі під час кожної епохи навчання. Приклади результатів наведені на рисунках 3 та 4.

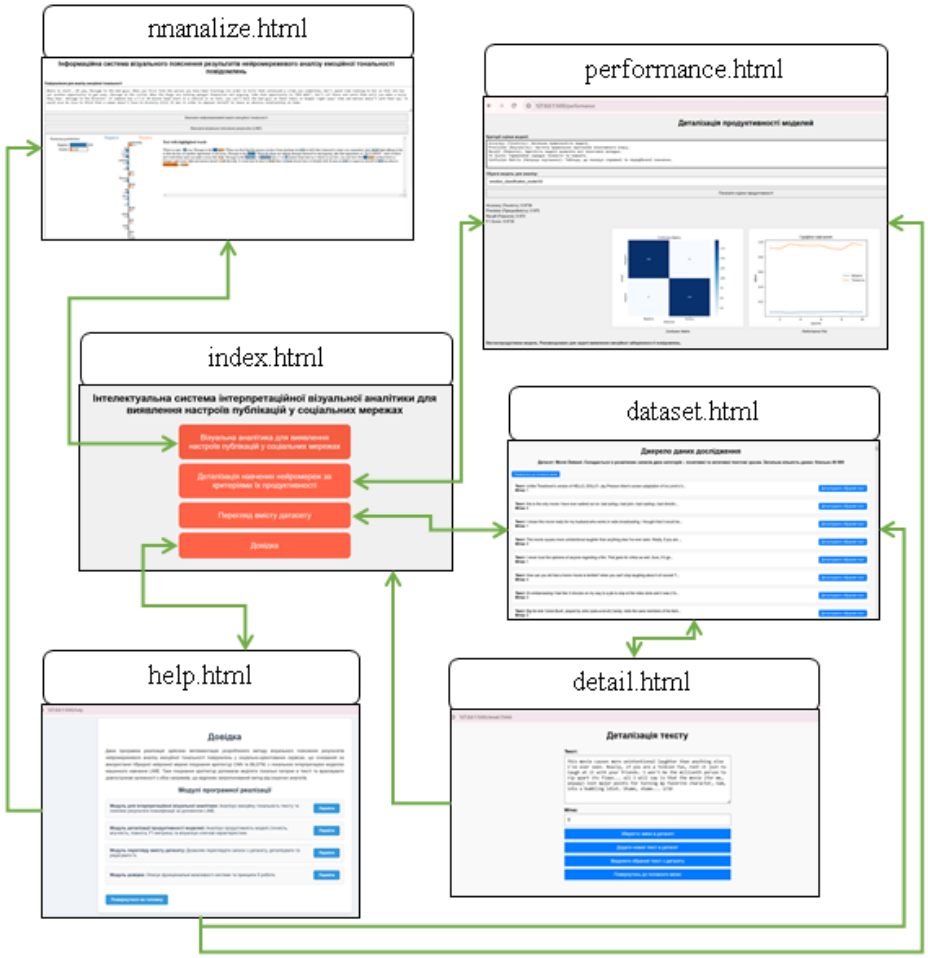


Рисунок 2 – Вебзастосунок для інтерпретаційної візуальної аналітики

```
Epoch 1/10
2000/2000 [==============================] - 159s 78ms/step - loss: 0.4396 - accuracy: 0.7858 - val_loss: 0.3343 - val_accuracy: 0.8528
Epoch 2/10
2000/2000 [==============================] - 156s 78ms/step - loss: 0.2725 - accuracy: 0.8866 - val_loss: 0.2770 - val_accuracy: 0.8830
Epoch 3/10
2000/2000 [==============================] - 156s 78ms/step - loss: 0.2312 - accuracy: 0.9064 - val_loss: 0.2749 - val_accuracy: 0.8829
Epoch 4/10
2000/2000 [==============================] - 152s 76ms/step - loss: 0.1952 - accuracy: 0.9227 - val_loss: 0.2914 - val_accuracy: 0.8826
Epoch 5/10
2000/2000 [==============================] - 155s 78ms/step - loss: 0.1644 - accuracy: 0.9355 - val_loss: 0.3170 - val_accuracy: 0.8765
Epoch 6/10
```

Рисунок 3 – Фрагмент навчання нейромережі гібридної архутектури

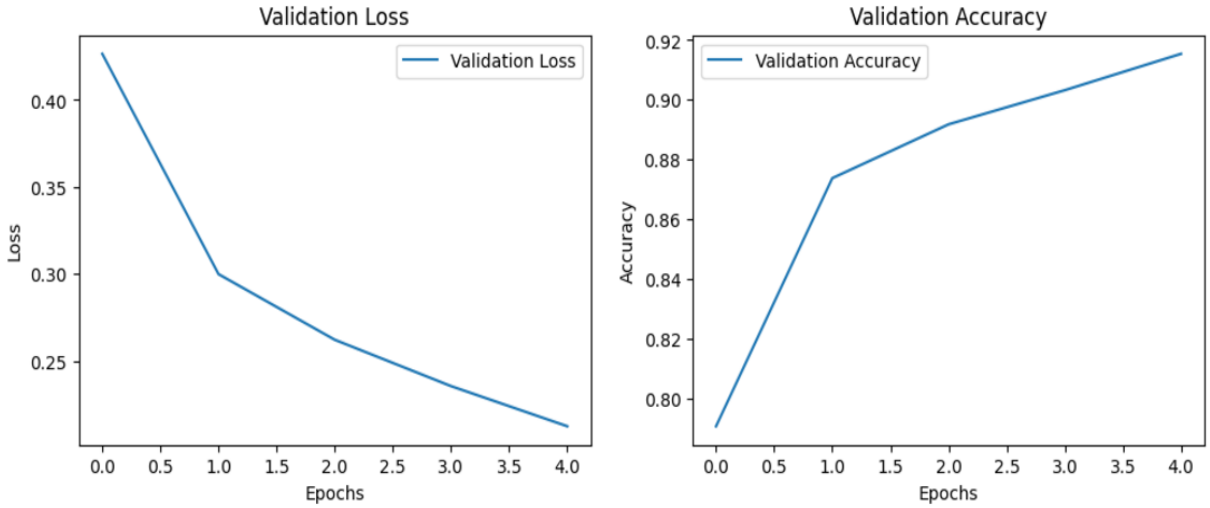


Рисунок 4 – Графіки навчання нейромережі

Навчена модель гібридної архітектури зберігається у файловій системі для подальшого використання у вебзастосунку з графічним інтерфейсом користувача. Запропонована програмна реалізація демонструє можливості використання методів машинного та глибокого навчання для автоматизованого виявлення маніпулятивних технік у текстовому контенті та забезпечення візуального пояснення результатів роботи нейромережевих моделей. Система охоплює повний цикл обробки даних - від підготовки текстового контенту та навчання моделей до інтерпретації результатів і їх візуального представлення користувачу.

Висновки. Отже, запропоновано метод, що базується на використанні гібридної нейромережевої архітектури CNN та BiLSTM у поєднанні з моделлю локальної інтерпретації LIME. Використання CNN забезпечує ефективне виявлення локальних текстових патернів і ключових мовних конструкцій, тоді як BiLSTM дозволяє враховувати контекстні залежності та семантичні особливості тексту. Застосування LIME дало можливість реалізувати механізм візуального пояснення результатів класифікації та визначити текстові елементи, що найбільше впливають на рішення моделі.

У межах дослідження реалізовано програмні компоненти інформаційної системи, які забезпечують аналіз текстового контенту, керування наборами даних, оцінювання ефективності нейромережевої моделі та формування візуальних аналітичних представлень результатів. Розроблений вебзастосунок забезпечує зручну взаємодію користувача із системою та підтримує функції інтерпретаційної візуальної аналітики. Отримані результати підтверджують доцільність використання методів глибокого навчання та засобів поясненного штучного інтелекту для підвищення прозорості нейромережевого аналізу текстового контенту. Запропонований підхід може бути використаний у системах моніторингу інформаційного простору, аналізу соціальних мереж, автоматизованої модерації контенту та підтримки прийняття рішень у сфері інформаційної безпеки. Перспективою подальших досліджень є підвищення точності виявлення маніпулятивних технік, розширення набору підтримуваних мов і вдосконалення методів інтерактивної візуалізації результатів роботи нейромережевих моделей.

Список літератури:

1. Chowdhury K. R., Sil A., Shukla S. R. Explaining a black-box sentiment analysis model with local interpretable model diagnostics explanation (LIME). *Advances in Computing and Data Sciences: 5th International Conference, ICACDS 2021*, Nashik, India, April 23–24, 2021, Revised Selected Papers, Part I 5, pp. 90-101.

2. Li Y., Chan J., Peko G., Sundaram D. An explanation framework and method for AI-based text emotion analysis and visualisation. *Decision Support Systems*, 178, 2024, p. 114121.

3. Krak, I., Yurchenko, D., Mazurets, O., Zalutska, O., Molchanova, M., Barmak, O. Method of Interpretive Visual Analytics for Neural Network Detection of Social Media Posts Sentiments. *Lecture Notes in Networks and Systems*. Springer, Cham. 2025. Vol. 1473. pp. 54–65.

Березька К.М. (1), Ілюшин В. В. (2)
1) доцент кафедри ПМ ФКІТ ЗУНУ, 2) магістрант 1 курсу ФКІТ ЗУНУ

МОДЕЛЮВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ АНАЛІЗУ ТРАНСПОРТНИХ ПОТОКІВ М. ТЕРНОПОЛЯ

Вступ. Зростання інтенсивності транспортних потоків у містах зумовлює необхідність удосконалення підходів до аналізу та прогнозування дорожнього руху. Для м. Тернополя актуальним є дослідження завантаженості ключових перехресть, оскільки саме вони формують основні вузли транспортної мережі та впливають на пропускну здатність вулично-дорожньої системи.

Аналіз сучасного стану транспортної інфраструктури м. Тернополя показує наявність проблем перевантаження окремих ділянок вулично-дорожньої мережі, що обумовлює необхідність використання методів транспортного моделювання для оцінювання інтенсивності руху та оптимізації транспортних потоків [1].

Ефективне планування транспортної інфраструктури міста потребує врахування не лише характеристик дорожньої мережі, але й соціально-економічних та демографічних факторів, що впливають на формування транспортних потоків [2].

Постановка задачі. Метою роботи є обґрунтування підходу до моделювання транспортних потоків м. Тернополя на основі даних інтенсивності руху транспортних засобів на окремих перехрестях та формування інформаційної основи для подальшого аналізу і прогнозування транспортного навантаження.

Для досягнення поставленої мети необхідно:

- проаналізувати структуру транспортних потоків на основних перехрестях м. Тернополя;
- сформувати інформаційну базу даних інтенсивності руху транспортних засобів;
- розробити підхід до представлення транспортних потоків у вигляді матриць ознак;
- обґрунтувати можливість використання графових, статистичних та кластерних моделей для аналізу транспортних потоків;
- розробити підхід до програмної реалізації аналізу транспортних потоків у середовищі R;
- визначити перспективні методи подальшого моделювання та прогнозування транспортного навантаження вулично-дорожньої мережі міста.

Об’єктом дослідження є транспортні потоки на вулично-дорожній мережі м. Тернополя. Предметом дослідження є методи аналізу та моделювання інтенсивності руху транспортних засобів на регульованих і нерегульованих перехрестях та їх програмна реалізація.

Для формування інформаційної бази дослідження можуть бути використані дані спостережень за інтенсивністю руху на основних транспортних вузлах міста. Зокрема, для кожного перехрестя доцільно фіксувати напрямки руху, кількість транспортних засобів за окремими потоками, часові інтервали спостереження, тип перехрестя, кількість смуг руху, наявність світлофорного регулювання та пішохідних переходів.

Використані дані. У роботі використовуються дані спостережень за інтенсивністю руху транспортних засобів на окремих перехрестях м. Тернополя. Дані подаються у вигляді схем транспортних потоків із зазначенням кількості транспортних засобів за напрямками руху протягом визначеного проміжку часу.

Для кожного транспортного вузла фіксуються:

- кількість транспортних засобів за напрямками руху;
- напрямки в’їзду та виїзду транспортних потоків;
- тип перехрестя;
- кількість смуг руху;
- наявність острівців безпеки та світлофорного регулювання;
- геометричні характеристики транспортного вузла.

Приклад схеми транспортних потоків на перехресті наведено на рис. 1.

На першому етапі дослідження передбачається систематизація первинних даних у вигляді матриць транспортних потоків. Такі матриці дозволять описати розподіл інтенсивності руху за

напрямками та визначити найбільш завантажені підходи до перехрестя. На наступному етапі планується застосування методів статистичного аналізу, кластеризації та імітаційного моделювання для виявлення типових режимів функціонування транспортних вузлів.

Перспективним є використання графової моделі вулично-дорожньої мережі, у якій вершинами виступають перехрестя, а ребрами – ділянки доріг між ними. Вагами ребер можуть бути інтенсивність руху, середня швидкість, пропускна здатність або умовний рівень завантаження. Такий підхід дає змогу формалізувати транспортну мережу міста та надалі використовувати алгоритми пошуку найкоротших шляхів, аналізу вузьких місць і прогнозування перевантажень.

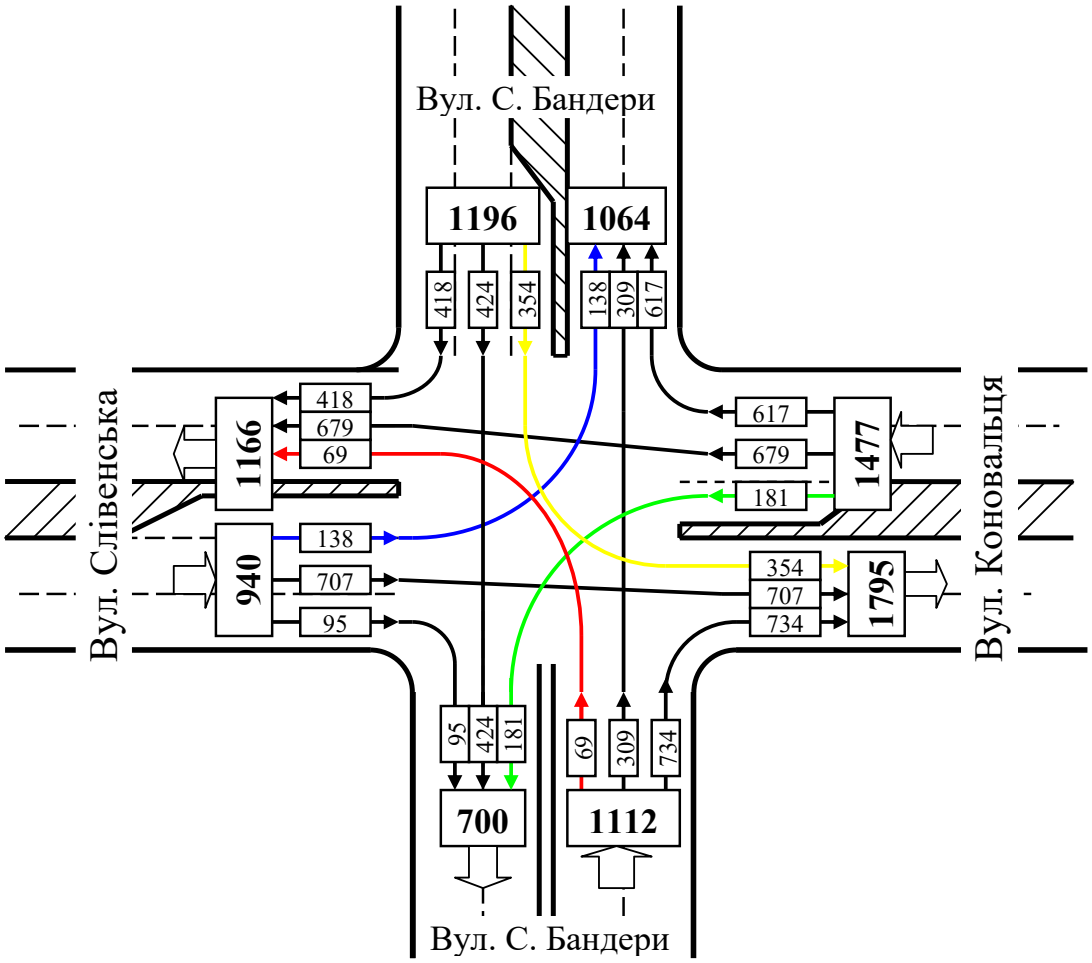


Рисунок 1. Приклад схеми транспортних потоків на перехресті м. Тернополя

Основний матеріал.

Для подальшого аналізу транспортні потоки можуть бути представлені у вигляді матриці ознак:

$$X = \{x_{ij}\}, i = 1,\ldots,n, j = 1,\ldots,m,$$

де n – кількість досліджуваних перехресть;

m – кількість характеристик транспортного потоку;

x_{ij} – значення j-ї ознаки для i-го перехрестя.

Як ознаки можуть використовуватися:

- інтенсивність руху;
- кількість смуг руху;
- середня швидкість транспортного потоку;
- рівень завантаженості перехрестя;

– кількість конфліктних точок;
– тип регулювання руху.

У межах дослідження кожне перехрестя розглядається як окремий транспортний вузол, що характеризується набором параметрів транспортного потоку. Це дозволяє застосовувати методи багатовимірного статистичного аналізу, кластеризації та графового моделювання.

Для формалізації транспортної мережі міста пропонується використання графової моделі:

$$G = (V, E),$$

де V – множина перехресть;

E – множина ділянок доріг між ними.

Вагами ребер графа можуть виступати інтенсивність руху, середня швидкість транспортного потоку або рівень транспортного навантаження.

Для обробки та аналізу транспортних даних передбачається розробка програмного модуля мовою R. Використання середовища R зумовлене наявністю широкого набору статистичних методів, засобів візуалізації даних та бібліотек для аналізу транспортних мереж і багатовимірних даних.

Програмний модуль повинен забезпечувати:
– завантаження та структуризацію даних транспортних потоків;
– формування матриць ознак транспортних вузлів;
– статистичний аналіз інтенсивності руху;
– кластеризацію транспортних вузлів за характеристиками завантаженості;
– побудову графової моделі транспортної мережі;
– візуалізацію транспортних потоків та результатів аналізу.

Для реалізації окремих етапів аналізу планується використання бібліотек dplyr, ggplot2, igraph, cluster та factoextra. Застосування цих засобів дозволить автоматизувати процес аналізу транспортних потоків та забезпечити подальше масштабування дослідження.

Висновки. У роботі обґрунтовано підхід до моделювання транспортних потоків м. Тернополя на основі даних інтенсивності руху на перехрестях. Запропоновано структуру представлення транспортних даних у вигляді матриць ознак та графової моделі вулично-дорожньої мережі.

Отримані результати та запропонований підхід можуть бути використані як основа для створення програмної системи аналізу транспортних потоків м. Тернополя та підтримки прийняття рішень у сфері організації дорожнього руху.

Подальші дослідження можуть бути спрямовані на розробку імітаційної моделі транспортних потоків, застосування методів кластерного аналізу та прогнозування транспортної завантаженості вулично-дорожньої мережі м. Тернополя.

Одним із перспективних напрямів є інтеграція транспортного моделювання з геоінформаційними системами для візуалізації транспортних потоків та аналізу просторових закономірностей руху.

Список літератури

1. Попович П.В., Маяк М.М., Розум Р.І., Буряк М.В., Березька К.М., Коваль Ю.Б., Мишко С.А. Дослідження стану транспортної інфраструктури міста Тернополя // Центральноукраїнський науковий вісник. Технічні науки. 2023. Вип. 7(38), Ч. II. С. 243–249. DOI: https://doi.org/10.32515/2664-262X.2023.7(38).2.243-249

2. Фалович В., Шевчук О., Фалович Н., Навольська Н., Захарчук В. Планування транспортної інфраструктури міст з урахуванням соціально-економічних та демографічних індикаторів розселення // Сучасні технології в машинобудуванні та транспорті. 2024. №1(22). С. 320–329. DOI: https://doi.org/10.36910/automash.v1i22.1375

Дребот С. Р.
студентка 4-того курсу ІКНІ НУ «Львівська Політехніка»
Науковий керівник доц. каф. АСУ Дорошенко А. В., каф. АСУ НУ «Львівська Політехніка»

ІНТЕЛЕКТУАЛЬНИЙ ПОМІЧНИК ІНСТРУКТОРА АВТОШКОЛИ НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ

Вступ. Забезпечення безпеки дорожнього руху є одним із найбільш актуальних завдань сучасного суспільства. За даними Всесвітньої організації охорони здоров'я, щороку внаслідок дорожньо-транспортних пригод гине понад 1,19 мільйона осіб, причому понад 90% аварій спричинені помилками водія [1]. Підготовка кваліфікованих водіїв в автошколах є першим і найважливішим етапом зниження аварійності. Проте, традиційна модель навчання має суттєве обмеження: інструктор не здатний одночасно стежити за дорогою, коригувати дії учня та об'єктивно фіксувати всі порушення. Суб'єктивність оцінювання, пропуск помилок і відсутність структурованої статистики прогресу знижують ефективність навчального процесу. З розвитком технологій сучасні методи комп'ютерного зору та машинного навчання відкривають нові можливості для вирішення цієї проблеми. Автоматизований аналіз відеопотоку дозволяє в режимі реального часу фіксувати небезпечні ситуації, дозволяючи інструктору більш ефективно виконувати свою роботу та об'єктивно оцінювати знання учня-водія.

Постановка задачі. Об'єктом проєктування є інтелектуальна система для автоматизованого аналізу поведінки учня-водія на основі відеопотоку з відеореєстратора транспортного засобу в режимі наближеному до реального часу. Предметом дослідження є методи комп'ютерного зору та машинного навчання для детекції, трекінгу та класифікації дорожніх ситуацій, які трапляються під час водіння. Метою є розробка інтелектуальної системи, що виявляє порушення водіння у реальному часі та надає інструктору структуровану інформацію про якість керування учня-водія під час самої поїздки. Серед основних функцій системи можна виокремити такі: приймати відеопотік від стандартного відеореєстратора, який підтримує передачу відеопотоків за допомогою підтримки фай-вай; виявляти та відстежувати об'єкти дорожнього середовища між кадрами на відео; класифікувати небезпечні ситуації, для прикладу такі як різке гальмування, різке маневрування, недотримання дистанції; зберігати дані кожної навчальної сесії у відповідній базі даних, а також миттєво сповіщати інструктора при виявленні порушення доповідаючи йому про саме порушення, час на яку воно відбулося та інші дані зображено на рисунку 1. Ключовою вимогою є те, щоб затримка між отриманням кадру та виявленням порушення не більше двох секунд.

Основний матеріал. Запропонована система побудована за клієнт-серверною архітектурою та складається з чотирьох взаємопов'язаних підсистем. Відеореєстратор передає JPEG-кадри через WebSocket зі швидкістю 25 кадрів/с на асинхронний FastAPI-сервер, який координує роботу машинного навчання у пулі потоків без блокування основного циклу подій. Конвеєр машинного навчання послідовно виконує чотири етапи обробки: детекцію об'єктів, трекінг, обчислення просторово-часових ознак та класифікацію поведінки. При виявленні порушення підсистема виводу зберігає подію у хмарній базі даних та надсилає сповіщення на пристрій інструктора.

Для детекції об'єктів дорожнього середовища обрано YOLOv8n [2] – однопрохідний детектор на основі архітектури CSPDarknet та PAN-FPN, що забезпечує швидкість обробки 69.4 FPS при затримці 14.4 мс на кадр. Така продуктивність задовольняє жорстку вимогу реального часу та дозволяє виявляти релевантних учасників дорожнього руху без надмірних хибних спрацювань. Модель попередньо навчена на наборі даних COCO, що містить усі необхідні для системи класи об'єктів.

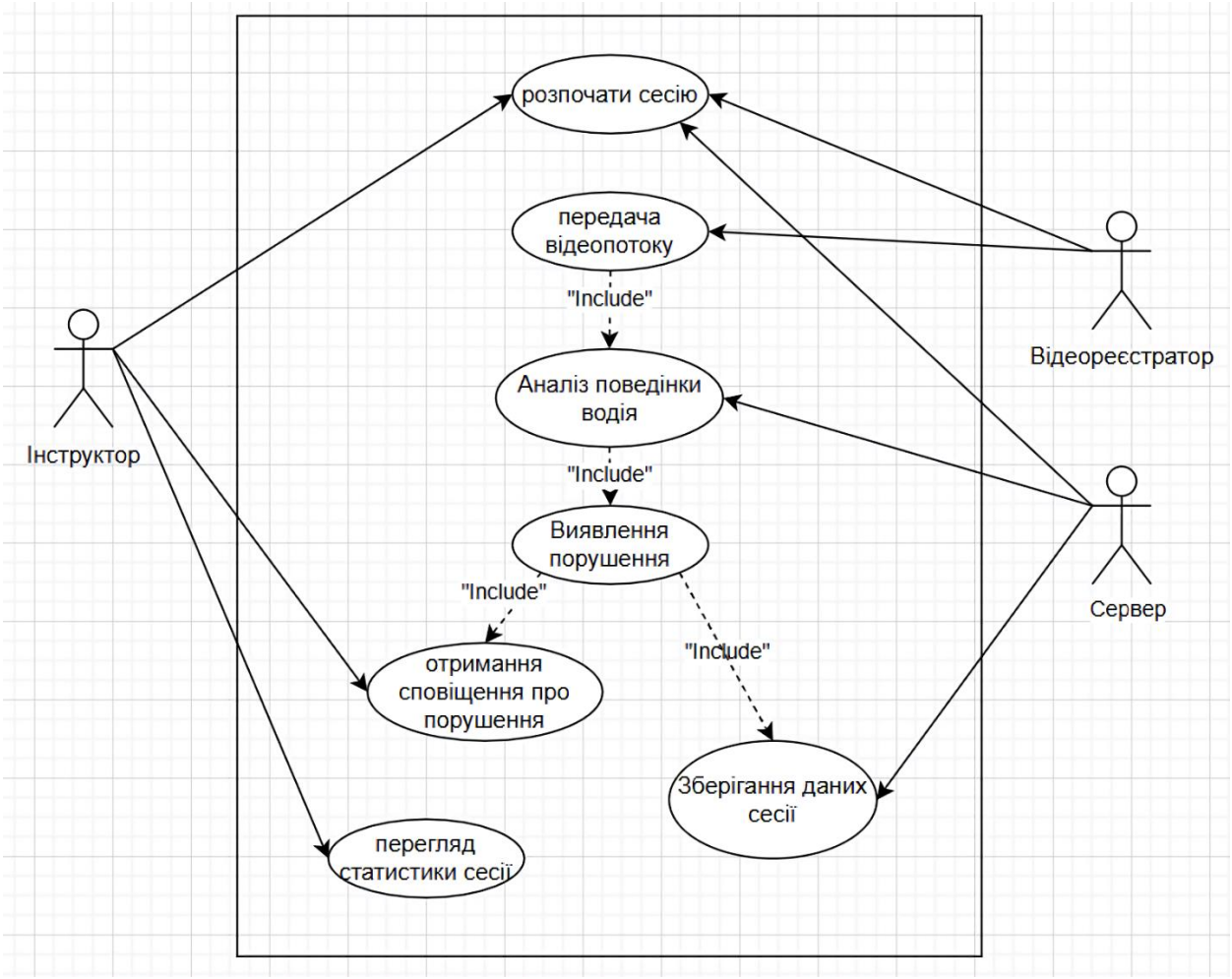


Рисунок 1 – Діаграма прецедентів (Use Case Diagram) системи

Для безперервного трекінгу виявлених об’єктів між кадрами застосовано DeepSORT [3] – алгоритм, що поєднує фільтр Калмана для прогнозування траєкторій із глибокою нейронною мережею обчислення ознак зовнішнього вигляду. Це дозволяє зберігати ідентичність об’єктів навіть при їх тимчасовому перекритті, що є необхідною умовою для коректного обчислення динамічних характеристик: швидкості, прискорення та відстані між транспортними засобами.

На основі обчислених просторово-часових ознак класифікатор поведінки приймає рішення про наявність порушення. Для класифікації застосовується модель XGBoost [4], попередньо навчена на відповідному наборі даних – масштабній базі відеоданих дорожнього руху з розміченими ситуаціями водіння. Виявлені порушення зберігаються у хмарній базі даних – для кожної події фіксуються тип, текстовий опис та часова мітка у відео – і водночас доставляються на пристрій інструктора. Взаємодію компонентів зображено на рисунку 2.

Практична цінність системи полягає у тому, що вона не потребує спеціалізованого обладнання: достатньо стандартного відеореєстратора з підтримкою Wi-Fi. Інструктор отримує структурований звіт кожного заняття з детальним описом порушень та часовими мітками, що суттєво підвищує об’єктивність оцінювання та якість підготовки водіїв.

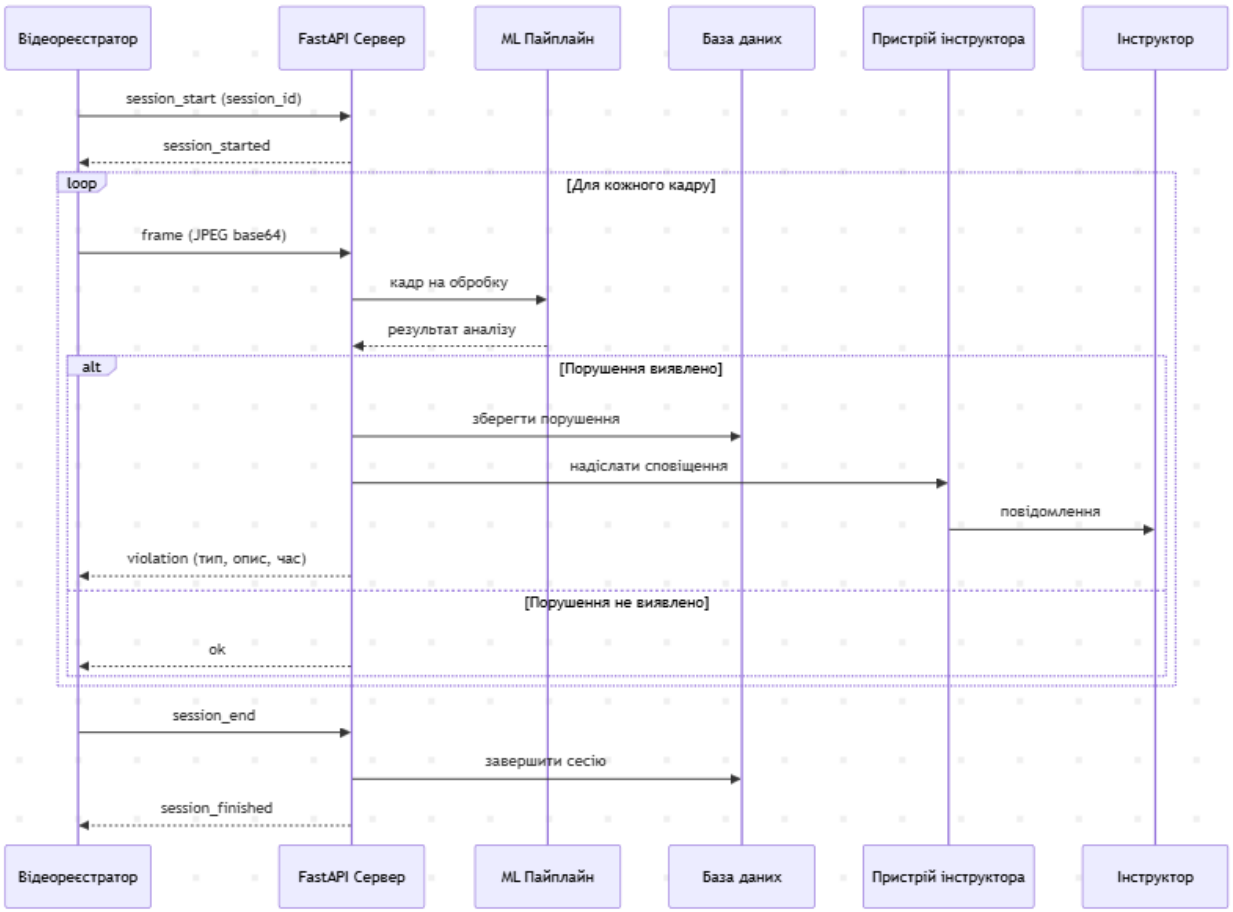


Рисунок 2 – Діаграма послідовності: взаємодія компонентів при виявленні порушення.

Висновки. Запропоновано концепцію інтелектуальної системи аналізу поведінки учня-водія на основі комп'ютерного зору. Система вирішує актуальну проблему суб'єктивності оцінювання в автошколах шляхом автоматизованого аналізу відеопотоку з відеореєстратора в режимі реального часу. Для детекції об'єктів дорожнього середовища обрано YOLOv8n, для трекінгу – DeepSORT, для класифікації поведінки – XGBoost. Запропонована клієнт-серверна архітектура з послідовним ML-пайплайном забезпечує виявлення порушень із затримкою менше двох секунд та оперативну доставку сповіщень на пристрій інструктора. Перевагою рішення є відсутність вимог до спеціалізованого обладнання – достатньо стандартного відеореєстратора з підтримкою Wi-Fi, що робить систему доступною для будь-якої автошколи.

Список літератури

1. 1.World Health Organization. Global status report on road safety 2023. – Geneva : WHO, 2023. – Режим доступу: https://www.who.int/publications/i/item/9789240086517

2. Jocher G. et al. Ultralytics YOLO. – 2023. – Режим доступу: https://github.com/ultralytics/ultralytics

3. Du Y., Zhao Z., Song Y., Zhao Y., Su F., Gong T., Meng H. StrongSORT: Make DeepSORT Great Again // IEEE Transactions on Multimedia. – 2023. – Vol. 25. – P. 8725–8737. DOI: 10.1109/TMM.2023.3240881

4. Navratil G., Giannopoulos I. Classifying Motorcyclist Behaviour with XGBoost Based on IMU Data // Sensors. – 2024. – Vol. 24, No. 3. – P. 1042. DOI: 10.3390/s24031042

Ілюшин В. В. (1), Король В.Р.(2)
1) магістрант 1 курсу ФКІТ ЗУНУ, 2) студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н Березький О. М., кафедра КІ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ КЛАСТЕРИЗАЦІЇ ПАСАЖИРОПОТОКІВ МІСТА ТЕРНОПОЛЯ

Вступ. Сучасні міські транспортні системи характеризуються складною структурою пасажиропотоків, нерівномірністю навантаження зупинок громадського транспорту та необхідністю оперативного прийняття управлінських рішень. Аналіз транспортної інфраструктури м. Тернополя засвідчує актуальність використання математичних і програмних методів для дослідження її функціонування [1, 4].

Проблеми безпеки дорожнього руху, інтенсивності транспортних потоків та оптимізації маршрутної мережі розглянуті у роботах [2, 5], де підкреслюється необхідність застосування сучасних методів обробки даних. Одним із ефективних інструментів аналізу є кластерний аналіз, що дозволяє виявляти групи об'єктів із подібними характеристиками.

У роботі [3] показано можливість застосування кластерного аналізу для визначення рівня завантаження зупинок громадського транспорту. Проте актуальною залишається задача створення програмного модуля, який забезпечує автоматизований процес кластеризації пасажиропотоків та оцінювання якості отриманих результатів.

Постановка задачі. Метою роботи є розробка програмного модуля кластеризації пасажиропотоків міста Тернополя на основі даних щодо середньої кількості пасажирів на зупинках у різні періоди доби (ранок, обід, вечір).

Для досягнення поставленої мети необхідно:

- здійснити формалізацію вхідних даних у вигляді векторів ознак;
- обґрунтувати вибір метрики подібності;
- реалізувати алгоритм кластеризації k-means;
- розробити механізм вибору оптимальної кількості кластерів;
- здійснити програмну реалізацію модуля;
- провести експериментальні дослідження та проаналізувати результати.

Об'єктом дослідження є пасажиропотоки міського громадського транспорту. Предметом дослідження є методи кластерного аналізу та їх програмна реалізація.

Основний матеріал. Вхідні дані подано у вигляді матриці ознак:

$$X = \{x_i\}_{i=1}^{n}, \quad x_i = \left(x_{i1}, x_{i2}, x_{i3}\right),$$

де n – кількість зупинок громадського транспорту;

x_i – вектор ознак i-ї зупинки;

x_{i1}, x_{i2}, x_{i3} – середня кількість пасажирів на зупинці i у ранковий, денний та вечірній періоди відповідно.

Таким чином, кожна зупинка подається як точка у тривимірному просторі ознак.

Перед кластеризацією виконано нормалізацію даних методом Min–Max:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}},$$

де x – початкове значення ознаки;

$x_{\min}$ – мінімальне значення ознаки у вибірці;

$x_{\max}$ – максимальне значення ознаки у вибірці;

x' – нормалізоване значення ознаки.

Нормалізація дозволяє привести всі ознаки до інтервалу [0;1], що запобігає домінуванню змінних з більшими числовими значеннями при обчисленні відстаней.

Для групування зупинок застосовано алгоритм k-means, що мінімізує функцію:

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} \| x_i - \mu_j \|^2,$$

де k – кількість кластерів;

C_j – множина об'єктів, що належать до j-го кластера;

μ_j – центр кластера C_j;

$\| x_i - \mu_j \|^2$ – квадрат евклідової відстані між об'єктом та центром кластера.

Вибір кількості кластерів здійснювався за методом "elbow" (аналіз WCSS) та коефіцієнтом силуету:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))},$$

де $a(i)$ – середня відстань між об'єктом iii та іншими об'єктами його кластера;

$b(i)$ – мінімальна середня відстань між об'єктом iii та об'єктами найближчого сусіднього кластера;

$S(i) \in [-1;1]$.

Програмний модуль реалізовано у вигляді структурованого Python-застосунку, що складається з окремих функціональних блоків: модуля зчитування даних, блоку попередньої обробки та нормалізації, блоку кластеризації, підсистеми оцінювання якості та блоку візуалізації результатів. Така модульна архітектура забезпечує гнучкість, масштабованість і можливість подальшого розширення функціоналу.

Зчитування даних здійснюється за допомогою бібліотеки pandas, що дозволяє працювати з табличними структурами даних та виконувати первинну перевірку коректності вхідної інформації. Нормалізація ознак реалізована з використанням інструментів scikit-learn (MinMaxScaler), що забезпечує приведення показників до єдиного масштабу.

Алгоритм k-means реалізовано засобами модуля sklearn.cluster, що дозволило скористатися оптимізованими реалізаціями числових методів. Для автоматизованого підбору кількості кластерів використано цикл перебору параметра k із розрахунком значень WCSS та коефіцієнта силуету. Результати обчислень зберігаються у структурованому форматі та можуть бути використані для подальшого аналізу.

Візуалізація результатів кластеризації здійснюється за допомогою бібліотеки matplotlib, що дозволяє будувати графіки типу "elbow", діаграми силуету та двовимірні проєкції кластерів. Реалізовано можливість експорту результатів у формат CSV, що забезпечує інтеграцію модуля з іншими аналітичними системами.

З позицій інтелектуальних комп'ютерних систем розроблений програмний модуль можна розглядати як компонент системи підтримки прийняття рішень у сфері транспортного планування. Модуль реалізує механізм адаптивного вибору параметрів кластеризації на основі критеріїв якості, що відповідає принципам самоналаштовуваних інтелектуальних систем. Використання методів машинного навчання дозволяє автоматично виявляти приховані закономірності у структурі пасажиропотоків без попереднього формулювання жорстких правил. Таким чином, програмний модуль формує основу для побудови інтелектуальної аналітичної платформи, здатної до масштабування, інтеграції з інформаційними мережами та подальшого розширення функціональності.

У результаті експериментальних досліджень виділено 8 кластерів, що відповідають різним типам зупинок: центральні вузли з високим навантаженням, житлові райони, навчальні заклади, периферійні зони тощо. Отримані результати узгоджуються з попередніми дослідженнями транспортної інфраструктури міста [1–4].

На рисунку 1 приведені середні профілі "ранок–обід–вечір" для кожного кластера (стовпчиковий графік).

Розроблений модуль реалізує повний програмний пайплайн: завантаження даних, нормалізація, підбір параметра k, кластеризація, оцінювання якості та візуалізація результатів.

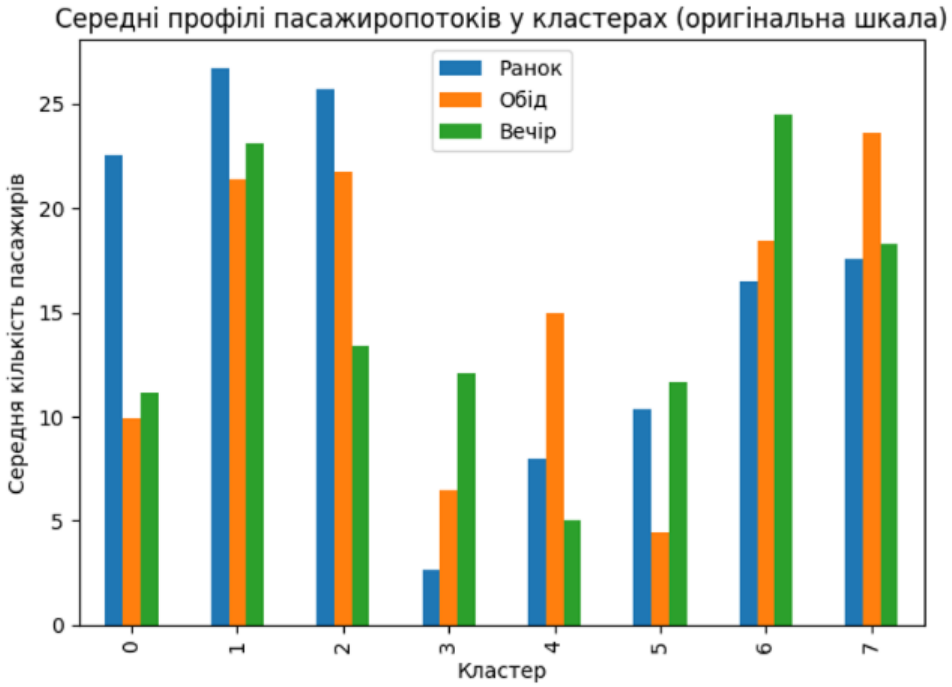


Рисунок 1 – Середні профілі “ранок–обід–вечір” для кожного кластера

Запропонований програмний модуль може бути інтегрований у ширші інформаційно-аналітичні системи підтримки прийняття рішень у сфері міського транспортного планування.

Висновки. У роботі розроблено програмний модуль кластеризації пасажиропотоків міста Тернополя. Реалізовано алгоритм k-means із механізмом вибору оптимальної кількості кластерів. Проведені експериментальні дослідження підтвердили ефективність запропонованого підходу для аналізу структури пасажиропотоків. Отримані результати можуть бути використані для оптимізації маршрутної мережі, коригування розкладів руху та модернізації транспортної інфраструктури середніх міст України. Подальші дослідження можуть бути спрямовані на використання часових рядів, інтеграцію з геоінформаційними системами та застосування альтернативних алгоритмів кластеризації.

Список літератури

1. Попович П.В., Маяк М.М., Розум Р.І., Буряк М.В., Березька К.М., Коваль Ю.Б., Мишко С.А. Дослідження стану транспортної інфраструктури міста Тернополя // Центральноукраїнський науковий вісник. Технічні науки. 2023. Вип. 7(38), Ч. II. С. 243–249. DOI: https://doi.org/10.32515/2664-262X.2023.7(38).2.243-249

2. Попович П.В., Розум Р.І., Мурований І.С., Буряк М.В., Березька К.М., Петринюк Н.А., Лоїк І.О. Дослідження безпеки дорожнього руху у м. Тернополі // Центральноукраїнський науковий вісник. Технічні науки. 2023. Вип. 7(38), Ч. II. С. 250–256. DOI: https://doi.org/10.32515/2664-262X.2023.7(38).2.250-256

3. Шевчук О.С., Березька К.М., Фалович Н.М., Захарчук О.П., Фалович В.А., Мамрош І.М. Визначення рівня завантаження зупинок громадського транспорту на основі кластерного аналізу // Сучасні технології в машинобудуванні та транспорті. 2024. №1(22). С. 357–368. DOI: https://doi.org/10.36910/automash.v1i22.1379

4. Шевчук О.С., Захарчук О.П., Фалович Н.М., Березька К.М., Сіран Р.В. Концептуальні основи модернізації транспортної інфраструктури середніх міст в Україні // Сучасні технології в машинобудуванні та транспорті. 2024. №1(22). С. 369–377. DOI: https://doi.org/10.36910/automash.v1i22.1380

5. Березька К.М., Шевчук О.С., Фалович Н.М., Бубняк Ю.Р. Аналіз проблем і математичних методів для їх вирішення в транспортній логістиці // Центральноукраїнський науковий вісник. Технічні науки. 2024. Вип. 9(40), Ч. II. С. 174–181. DOI: https://doi.org/10.32515/2664-262X.2024.9(40).2.174-181

Чаус М.В.
студентка 4 курсу, кафедра АСУ, ІКНІ, НУ Львівська Політехніка
Науковий керівник: асистент Павенська А.-М. В., кафедра АСУ, НУ Львівська Політехніка

СМАРТ-СИСТЕМА ДЛЯ ВИВЧЕННЯ ЖЕСТОВОЇ МОВИ З АДАПТИВНИМ НАЛАШТУВАННЯМ НАВЧАЛЬНОЇ ПРОГРАМИ ТА ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ І МАШИННОГО НАВЧАННЯ

Вступ. Жестові мови є природними людськими мовами з власною фонологією, морфологією та синтаксисом, що розвиваються всередині спільнот глухих людей у всьому світі. Українська жестова мова (УЖМ) є мовою спілкування спільноти глухих в Україні, правовий статус та використання якої регулюється низкою законодавчих актів, зокрема Законом України " Про забезпечення функціонування української мови як державної", та слугує засобом комунікації для приблизно 50 тисяч осіб з порушеннями слуху. Однак доступ до якісної освіти УЖМ залишається критично обмеженим, менше 2% з 34 мільйонів глухих дітей у світі отримують доступ до жестової мови в ранньому дитинстві, що створює високий ризик мовної депривації - постійної відсутності доступу до природної мови в критичний період мовного розвитку впродовж перших п'яти років життя.[1] Цифрові ресурси для вивчення УЖМ залишаються вкрай обмеженими. Хоча в Україні існують ініціативи з підтримки інклюзивної освіти, такі як Всеукраїнська онлайн-школа, яка почала інтегрувати переклад навчальних матеріалів у жестову мову, комплексних інтерактивних платформ для самостійного вивчення УЖМ фактично не існує. Більшість існуючих ресурсів обмежуються відеословниками без інтерактивних вправ або механізмів закріплення знань.

Постановка задачі. Метою дослідження є створення веборієнтованої навчальної платформи для опанування української жестової мови, що інтегрує методи автоматичного розпізнавання динамічних жестів із педагогічними підходами адаптивного навчання. Об'єктом дослідження виступає процес навчання жестовим мовам з використанням технологій комп'ютерного зору та машинного навчання. Предмет дослідження - методи автоматичного розпізнавання динамічних жестів української жестової мови на основі аналізу ключових точок опорно-рухового апарату та їх інтеграція в адаптивну навчальну систему з механізмами інтервального повторення.

Основний матеріал. Система базується на клієнт-серверній архітектурі з повним розділенням фронтенду та бекенду. **Бекенд** реалізовано на Python з використанням RESTful API та реляційної бази даних PostgreSQL. Серверна частина організована модульно для обробки автентифікації, управління контентом (уроки, жести), відстеження прогресу користувачів та геймфікаційних елементів. База даних зберігає інформацію про користувачів, навчальний контент з багатомовною підтримкою, статистику виконання вправ та досягнення.

Фронтенд побудовано на React з підтримкою локалізації (українська/англійська) та адаптивними темами оформлення. Для розпізнавання жестів використовується бібліотека MediaPipe, що працює повністю в браузері користувача. **Логіка навчання** реалізована через адаптивну систему: після кожної вправи оновлюється статистика жестів, обчислюється рівень майстерності (0-3) та планується наступне повторення для закріплення матеріалу. Система автоматично генерує щоденні завдання, відстежує активність користувача, нараховує досвід (XP) та поступово відкриває новий контент відповідно до прогресу навчання.

Розпізнавання жестів базується на двоетапному підході: екстракція ключових точок з відеопотоку за допомогою MediaPipe та класифікація часових послідовностей рекурентною нейронною мережею. **MediaPipe Hands** визначає 21 ключову точку для кожної руки (зап'ястя, долоня, суглоби пальців). Положення тіла оцінюється за допомогою моделі **BlazePose**, реалізованої в MediaPipe, яка виявляє 33 точки скелету (плечі, лікті, зап'ястя, торс). Кожна точка представлена координатами (x, y, z), де x та y нормалізовані до [0, 1], а z - відносна глибина. [2] [3]

Навчальний датасет створено шляхом запису 80-100 варіантів виконання для кожного жесту у форматі 1920×1080 пікселів, 30 fps. Послідовності нормалізовані до 30 кадрів. Застосовано аугментацію: гаусівський шум, масштабування часу (±10-20%), горизонтальне дзеркалювання. Фінальний датасет: 1500 прикладів для 15 жестів УЖМ (80% навчання, 20% валідація).

Порівнювані архітектури

1. GRU (Gated Recurrent Unit) Базова рекурентна архітектура з трьома шарами GRU (64-128-64 одиниць) та двома dense шарами. GRU є спрощеною версією LSTM з меншою кількістю параметрів, що прискорює навчання та inference.

2. GRU з Attention Розширення базової GRU архітектури з додаванням механізму Multi-Head Attention (4 голови, key_dim=128) після другого GRU шару. Skip-connection та layer normalization забезпечують стабільність навчання глибшої мережі.

3. LSTM (Long Short-Term Memory) Класична рекурентна архітектура з трьома LSTM шарами (64-128-64 одиниць). LSTM має більше параметрів на комірку порівняно з GRU завдяки окремим вентилям (gates) для забування, вводу та виводу, що теоретично дозволяє краще моделювати довгострокові залежності.

4. LSTM з Attention Гібридна архітектура, що поєднує два шари LSTM з Multi-Head Attention механізмом. Архітектурно ідентична до GRU з Attention, але використовує LSTM замість GRU комірок.

5. Transformer[4] Сучасна архітектура без рекурентних шарів, заснована виключно на механізмах self-attention. Включає sinusoidal positional encoding для збереження інформації про порядок кадрів, два Transformer encoder блоки з Multi-Head Attention (4 голови, head_size=64) та feed-forward мережами (256 одиниць).

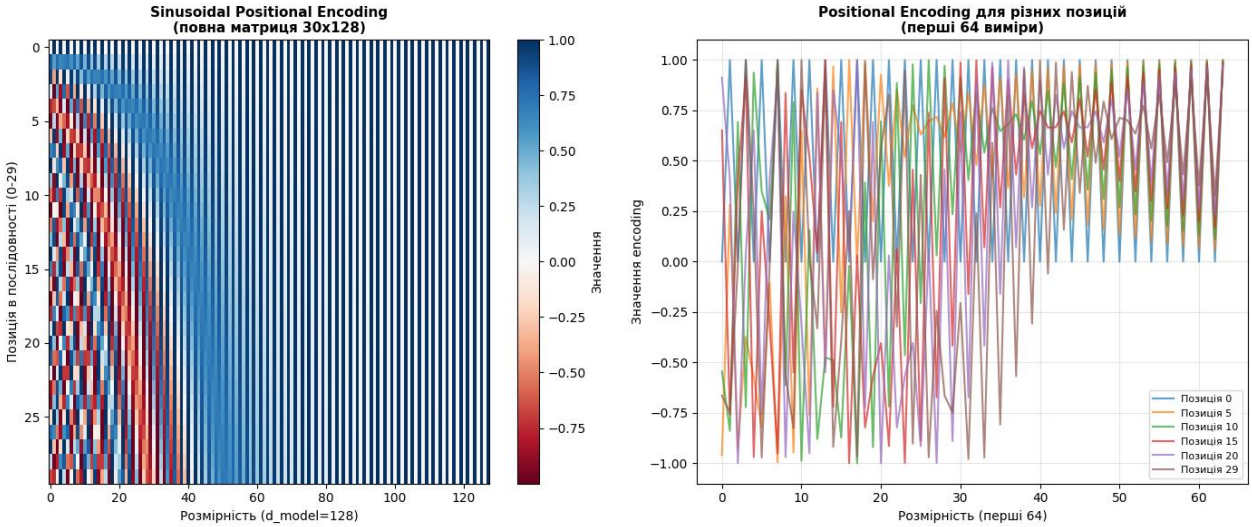


Рисунок 1 – Позиційне кодування для transformer

Параметри навчання: batch size 16, optimizer Adam, loss categorical cross-entropy, dropout 0.3-0.4.

Таблиця 1. Порівняння архітектур

Модель	Параметри	Точність (Val)	F1-score	Overfitting Gap	Час inference (мс)
GRU	217,551	84.9%	0.85	0.067	15-20
GRU + Attention	448,463	**89.3%**	0.89	0.029	25-30
LSTM	286,799	87.5%	0.88	0.024	18-23
LSTM + Attention	505,551	88.6%	0.88	**0.007**	28-35
Transformer	480,271	82.3%	0.83	0.049	30-40

Результати: GRU+Attention показала найвищу точність (89.3%). LSTM+Attention - найменший overfitting gap (0.007), найкращу генералізацію.

Висновки. У ході дослідження розроблено смарт-систему для вивчення української жестової мови на основі технологій комп'ютерного зору та машинного навчання. Проведено порівняльний аналіз п'яти архітектур нейронних мереж для розпізнавання жестів УЖМ. Експериментальні результати показали, що гібридна архітектура GRU з механізмом Multi-Head Attention демонструє найкращий баланс між точністю розпізнавання (89.3%) та швидкістю обробки (25-30 мс), що робить її оптимальним вибором для інтерактивної навчальної платформи реального часу. Подальші напрямки: розширення словника до 100+ жестів, інтеграція граматичних конструкцій УЖМ, тестування з користувачами-носіями мови.

Список літератури

1. World Report on Hearing. Geneva: World Health Organization, 2021. 250 p. URL: World Report on Hearing (дата звернення: 16.05.2026).

2. Zhang F., Bazarevsky V., Vakunov A., Tkachenka A., Sung G., Chang C.-L., Grundmann M. MediaPipe Hands: On-device Real-time Hand Tracking // Proceedings of the CV4ARVR Workshop at CVPR. 2020. URL: MediaPipe Hands paper (arXiv) (дата звернення: 16.05.2026).

3. Bazarevsky V., Grishchenko I., Raveendran K., Zhu T., Zhang F., Grundmann M. BlazePose: On-device Real-time Body Pose Tracking. 2020. URL: BlazePose paper (arXiv) (дата звернення: 16.05.2026).

4. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention Is All You Need // Advances in Neural Information Processing Systems. 2017. Vol. 30. URL: Attention Is All You Need (arXiv) (дата звернення: 16.05.2026).

Ткачук Катерина Ігорівна
Аспірант 3 року навчання, каф. АСУ, ІКНІ, НУЛП
Науковий керівник к.т.н., доцент Кустра Н.О.

РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ У ПОВЕДІНКОВИХ ДАНИХ СТУДЕНТІВ НА ОСНОВІ АВТОЕНКОДЕРА

Вступ. У сучасних освітніх системах накопичуються великі обсяги поведінкових та академічних даних студентів, аналіз яких дозволяє виявляти фактори, що впливають на успішність навчання та поведінку студентів. У наукових дослідженнях зростає інтерес до застосування методів машинного та глибокого навчання для аналізу освітніх даних і виявлення аномалій, що можуть сигналізувати про потенційні проблеми в навчальному процесі. Аналіз сучасних наукових публікацій свідчить про стрімкий розвиток освітньої аналітики. Зокрема, у працях дослідників розглядаються архітектури на основі згорткових автоенкодерів та нейронних мереж LSTM для прогнозування відрахування студентів [1], інші роботи фокусуються на сучасних підходах до освітньої аналітики та задачі виявлення аномалій у поведінкових даних [2]. Дослідження публікацій показало, що традиційні методи, зокрема статистичні підходи, Isolation Forest та One-Class SVM, мають обмеження при роботі з високовимірними та нелінійними даними. У зв'язку з цим актуальною є розробка системи виявлення аномалій на основі автоенкодера, який здатний ефективно моделювати складні залежності та виявляти приховані відхилення у поведінкових характеристиках студентів.

Постановка задачі. Об'єктом дослідження є поведінкові та академічні дані студентів, що характеризують їхню навчальну діяльність та особливості поведінки. Предметом дослідження є методи та алгоритми виявлення аномалій у поведінкових даних із використанням автоенкодерів. Метою роботи є розробка системи виявлення аномалій у поведінкових даних студентів на основі методів глибокого навчання з використанням бібліотеки PyTorch.

Для досягнення поставленої мети необхідно:

- провести аналіз існуючих методів виявлення аномалій;
- виконати передобробку та нормалізацію даних;
- розробити архітектуру автоенкодера;
- реалізувати алгоритм навчання моделі;
- визначити критерій виявлення аномалій;
- оцінити ефективність моделі за допомогою метрик якості.

Основний матеріал. Аномалії - це спостереження, які значно відрізняються від більшості даних у наборі. У контексті поведінкових даних студентів аномалії можуть вказувати на нетипові моделі поведінки, які потребують уваги з боку викладачів чи адміністрації. Наприклад, студент, який раніше регулярно відвідував заняття, але раптово почав пропускати їх, може бути позначений як аномалія.

Автоенкодер - це тип нейронної мережі, що складається з двох основних частин: енкодера, який стискає вхідні дані до простору меншої розмірності (латентний простір), та декодера, який відновлює дані з латентного простору. Основна ідея використання автоенкодера для виявлення аномалій полягає в тому, що модель навчається реконструювати нормальні дані з мінімальною помилкою. Аномалії, які мають відмінні від нормальних даних характеристики, матимуть більшу помилку реконструкції. Математично, якщо вхідні дані позначено як x, енкодер відображає їх у латентний простір $z = f(x)$, а декодер намагається відновити дані $\bar{x} = g(z)$.

Помилка реконструкції обчислюється як середньоквадратична помилка (MSE). Дані з високою помилкою реконструкції класифікуються як аномалії.

Для дослідження використано датасет student_habits_performance.csv, що містить поведінкові та академічні характеристики студентів: кількість годин навчання, тривалість сну, участь у позакласних заходах, наявність роботи за сумісництвом та середній бал.

Перед навчанням моделі проведено передобробку даних, яка включала видалення нерелевантних ознак, кодування бінарних параметрів, перетворення даних до числового

формату, видалення пропущених значень та масштабування даних у діапазоні [0;1] за допомогою методу MinMaxScaler.

Для виявлення аномалій реалізовано автоенкодер, який складається з енкодера та декодера. Архітектура моделі включає:

• Енкодер: три шари (вхідний шар → 128 → 64 → 32 нейрони) з функцією активації ReLU та шаром Dropout (0.1) для зменшення перенавчання.

• Декодер: три шари (32 → 64 → 128 → вхідний шар) з ReLU та Dropout, завершальний шар використовує сигмоїдну функцію для відповідності масштабованим даним.

Така архітектура дозволяє стискати дані до 32-вимірного латентного простору, що забезпечує баланс між компресією та збереженням інформації. Навчання автоенкодера проводилося лише на нормальних даних із використанням функції втрат MSE:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2,$$

де n -кількість елементів, x_i- справжнє значення, $\bar{x}_i$-передбачене значення моделі.

Для оптимізації параметрів нейронної мережі використано алгоритм Adam з кроком навчання 0.001, максимальна кількість епох - 100. Для запобігання перенавчанню застосовано механізм ранньої зупинки, який припиняє навчання у випадку відсутності покращення значення функції втрат протягом 10 епох.Поріг аномалій визначено як 95-й процентиль помилок реконструкції на тренувальних даних:

$$threshold = percentile(MSE, 95)$$

Дані з помилкою реконструкції, що перевищує поріг, класифікуються як аномалії.

Оцінювання ефективності моделі проводилося за допомогою метрик Accuracy, Precision, Recall та F1-score. У результаті експериментів модель продемонструвала високі показники якості: Accuracy - 0.92–0.95, Precision - 0.88–0.93, Recall - 0.85–0.90, F1- score - 0.86–0.91. Для оцінки чутливості моделі проведено експерименти з різними значеннями гіперпараметрів:

• Зменшення розміру до 16 нейронів призводило до втрати інформації, тоді як збільшення до 64 нейронів не давало значного покращення.

• Значення кроку навчання 0.001 виявилося оптимальним; зменшення до 0.0001 уповільнювало навчання, а збільшення до 0.01 призводило до нестабільності.

• Використання 90-го або 99-го процентиля змінювало баланс між Precision та Recall.

Модель автоенкодера порівнювалася з методами на основі Isolation Forest та One-Class SVM. Автоенкодер показав кращі результати за F1- score завдяки здатності моделювати нелінійні залежності. Наприклад, ізоляційний ліс мав нижчу повноту (0.75–0.80) через складність обробки високовимірних даних

Висновки: У результаті роботи розроблено систему виявлення аномалій у поведінкових даних студентів на основі автоенкодера з використанням бібліотеки PyTorch. Використання глибокого навчання дозволило обробляти складні нелінійні залежності в даних. Аналіз чутливості та порівняння з іншими методами підтвердили переваги автоенкодера. Практичне значення роботи полягає у створенні системи автоматизованого моніторингу поведінкових даних студентів, яка може бути використана в освітніх закладах для своєчасного виявлення потенційних проблем та підвищення ефективності освітнього процесу. Отримані результати підтвердили ефективність запропонованого підходу та перспективність його подальшого вдосконалення.

Список літератури

1. Niu, K., Lu, G., Peng, X. et al. CNN autoencoders and LSTM-based reduced order model for student dropout prediction. Neural Comput & Applic 35, 22341–22357 (2023). https://doi.org/10.1007/s00521-023-08894-2

2. Guo, T., Bai, X., Tian, X., Firmin, S., & Xia, F. (2022). Educational anomaly analytics: Features, methods, and challenges. Frontiers in Big Data, Data Mining and Management, 4, Article 811840. https://doi.org/10.3389/fdata.2021.811840

3. Zhang, K., Liu, C., Zhang, J., Xiong, H., Xing, E., & Ye, J. (2017). Randomization or condensation?: Linear-cost matrix sketching via cascaded compression sampling. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 615–623. https://doi.org/10.1145/3097983.3098050

Боднар А.О.
студентка 4 курсу ІКНІ НУ Львівська Політехніка
Науковий керівник асистент Загвойський Р.Ю., кафедра АСУ НУ Львівська Політехніка

ВЕБ-СИСТЕМА АВТОМАТИЗОВАНОГО СТВОРЕННЯ РЕКЛАМНОГО ВІДЕОКОНТЕНТУ НА ОСНОВІ ГЕНЕРАТИВНИХ ШІ-МОДЕЛЕЙ

Вступ. Стрімкий розвиток цифрового маркетингу зумовив суттєве зростання попиту на відеоконтент як один із найефективніших інструментів просування товарів та послуг. За даними аналітичних досліджень у сфері AIGC (AI-generated content), більшість маркетингових кампаній в електронній комерції орієнтована саме на короткі рекламні відео для соціальних мереж і маркетплейсів [1]. Відеореклама забезпечує значно вищий рівень залучення аудиторії порівняно зі статичними форматами завдяки можливості динамічної демонстрації товару, формування емоційного зв'язку з глядачем та прискореного впливу на прийняття рішення про покупку.

Водночас процес створення якісної відеореклами залишається складним і ресурсомістким. Він охоплює кілька послідовних етапів: написання рекламного сценарію, підготовку фото- та відеоматеріалів, підбір диктора або моделі, озвучення тексту, монтаж усіх елементів та фінальний рендеринг. У більшості випадків це вимагає використання кількох різних спеціалізованих програм і сервісів, що суттєво збільшує витрати часу й коштів. Особливо гостро ця проблема проявляється для малого бізнесу, індивідуальних продавців та контент-мейкерів, які не мають доступу до професійних маркетингових команд.

Розвиток великих мовних моделей (LLM), систем синтезу мовлення та технологій генерації ШІ-аватарів відкрив нові можливості для автоматизації виробництва рекламного контенту [2]. Проте інтеграція цих компонентів у межах єдиної спеціалізованої системи, орієнтованої саме на електронну комерцію, залишається актуальною науково-практичною задачею. Існуючі платформи є здебільшого універсальними інструментами і не формують цілісного автоматизованого процесу типу «товар → готова реклама».

Постановка задачі. Об'єктом дослідження є процеси автоматизованого створення мультимедійного рекламного контенту на основі технологій штучного інтелекту. Предметом дослідження є методи та підходи до генерації сценаріїв, синтезу мовлення, створення ШІ-аватарів та інтеграції сервісів штучного інтелекту в межах єдиної інформаційної системи. Метою роботи є розроблення веб-системи автоматизованої генерації відеореклами товарів із застосуванням генеративних моделей штучного інтелекту, яка забезпечує повний цикл створення рекламного відеоконтенту від вхідних даних про товар до готового ролика.

Для досягнення поставленої мети необхідно вирішити такі завдання: проаналізувати існуючі платформи автоматизованого створення відеоконтенту та виявити їх обмеження; розробити архітектуру системи з інтеграцією зовнішніх ШІ-сервісів; спроєктувати структуру даних та механізми обміну між компонентами; реалізувати алгоритм наскрізної генерації рекламного відео на основі вхідних даних користувача.

Формально задача описується як функціональне перетворення:

$$F: D \rightarrow R,$$

де D – сукупність вхідних параметрів (інформація про товар, медіафайли, параметри генерації), а R – готовий рекламний відеоролик, що відповідає заданим вимогам.

Основний матеріал. Аналіз сучасних рішень у сфері автоматизованого створення відеоконтенту охопив такі платформи: Synthesia, HeyGen та Pictory [3–5]. Synthesia та HeyGen спеціалізуються на генерації відео з цифровими аватарами і підтримують синтез мовлення, проте вимагають від користувача самостійної підготовки тексту і не орієнтовані на товарну рекламу. Pictory автоматично перетворює текст у відео з використанням стокових матеріалів, однак не підтримує ШІ-аватари та не забезпечує повного циклу від опису товару до готового ролика. Жодна з розглянутих платформ не пропонує наскрізного спеціалізованого процесу для електронної комерції з підтримкою користувацьких медіафайлів.

Архітектура розробленої системи побудована за моделлю C4 і реалізована як веборієнтований застосунок. Клієнтська частина розроблена на базі Next.js та React і забезпечує покроковий інтерфейс користувача. Серверна логіка реалізована на платформі Convex, яка поєднує базу даних, серверні функції та механізми реалтайм-синхронізації стану між клієнтом і сервером. Для генерації рекламних сценаріїв використовується OpenRouter як проксі-рівень до великих мовних моделей, що забезпечує гнучкість у виборі постачальника LLM. Синтез відеоаватара реалізовано через сервіс Akool AI, а зберігання та оптимізація медіафайлів – через ImageKit. Авторизацію користувачів забезпечує Clerk. Взаємодія між усіма компонентами реалізована через REST API з використанням документованого протоколу HTTPS і формату JSON.

Процес роботи системи є лінійним і складається з таких етапів. Користувач вводить інформацію про товар у вигляді текстового опису або посилання на сторінку товару. На основі цих даних LLM генерує кілька варіантів рекламного сценарію, з яких користувач обирає найбільш релевантний. Далі користувач завантажує фото та відеоматеріали товару через ImageKit, обирає ШІ-аватара і голос для озвучування. Після підтвердження всіх параметрів система формує запит до Akool AI, який повертає згенерований відеоролик. Результат зберігається у Convex і стає доступним для перегляду або завантаження.

Ключові відмінності розробленої системи від розглянутих аналогів полягають у наступному:

– вузька спеціалізація на товарній рекламі в електронній комерції, на відміну від універсальних відеоредакторів;

– автоматична генерація кількох варіантів рекламного сценарію великою мовною моделлю без участі користувача;

– підтримка завантаження користувацьких медіафайлів – реальних фото та відео конкретного товару;

– повна інтеграція всіх етапів виробництва реклами – генерація тексту, вибір аватара, синтез голосу та рендеринг – в межах єдиної платформи без необхідності використання сторонніх сервісів.

Висновки. У результаті проведеного дослідження розроблено прототип веб-системи автоматизованого створення рекламного відеоконтенту для товарів електронної комерції на основі генеративних моделей штучного інтелекту. Проведений аналіз існуючих платформ підтвердив відсутність спеціалізованих рішень, що забезпечують наскрізний автоматизований процес від опису товару до готового рекламного ролика. Наукова новизна роботи полягає у розробці архітектурного підходу до побудови інтегрованої системи, що поєднує генеративні ШІ-моделі різних типів – LLM для генерації тексту, синтез мовлення та відеоаватари – у єдиному спеціалізованому процесі генерації товарної реклами.

Практична цінність результатів полягає у можливості застосування розробленої системи малим бізнесом, інтернет-магазинами, маркетинговими агентствами та контент-мейкерами для швидкого й економічного створення рекламних матеріалів без залучення професійних студій відеомонтажу та спеціалізованого програмного забезпечення. Подальший розвиток системи передбачає розширення підтримуваних форматів вихідного відео, додавання можливості мультимовної генерації сценаріїв та інтеграцію з популярними платформами електронної комерції для автоматичного отримання даних про товар.

Список літератури

1. Wu, J., Gan, W., Chen, Z., Wan, S., & Lin, H. (2023). AI-generated content (AIGC): A survey.
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., & Amodei, D. (2021). Language models are few-shot learners. *Advances in Neural Information Processing Systems*.
3. Synthesia. (б. д.). *Synthesia*. https://www.synthesia.io
4. HeyGen. (б. д.). *HeyGen*. https://www.heygen.com
5. Pictory. (б. д.). *Pictory*. https://pictory.ai

Керничний В.Р.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник: к.т.н., доцент Дубчак Л.О., кафедра КІ ЗУНУ

HDL-МОДЕЛЬ НИЗЬКОЧАСТОТНОЇ ФІЛЬТРАЦІЇ ЗОБРАЖЕННЯ

Вступ. Цифрова обробка зображень є ключовою частиною сучасних інформаційних технологій. У швидкодіючих системах зросла популярність апаратної реалізації алгоритмів через вимоги до обробки в реальному часі та мінімізації затримок. HDL-моделі дають змогу виконувати обробку у вигляді цифрових апаратних структур з детермінованою поведінкою [1, 2].

Постановка задачі. Об’єктом дослідження є процеси цифрової обробки візуальних даних. Предмет дослідження - HDL-модель низькочастотної фільтрації зображень. Метою роботи є розробка та моделювання структури апаратного фільтра, що забезпечує згладжування зображення та пригнічення шумів у потоковому режимі [3].

Основний матеріал. Програмна реалізація HDL-моделі низькочастотної фільтрації зображення базується на описі архітектури пристрою мовами Verilog або VHDL, де ключовим елементом є модуль віконної обробки (sliding window). Процес починається з організації буферів рядків (Line Buffers), які дозволяють одночасно звертатися до декількох сусідніх пікселів для формування ядра згортки, наприклад, розміром 3×3. На кожному такті системного годинника дані зсуваються в регістрах, утворюючи матрицю, що множиться на коефіцієнти фільтра (наприклад, фільтр Гаусса або середнього арифметичного). Обчислювальна логіка реалізується через конвеєрну структуру (pipelining) для мінімізації критичного шляху: спочатку виконується паралельне множення значень пікселів на ваги, потім ієрархічне додавання отриманих добутків і, зрештою, нормування результату шляхом зсуву бітів або ділення.

Для перевірки працездатності розробляється Testbench на мові SystemVerilog, який зчитує вхідний файл зображення (BMP або RAW), подає його на вхід моделі у вигляді потоку пікселів та записує вихідний потік у новий файл для візуального порівняння результатів фільтрації. Синтез коду здійснюється в середовищах на кшталт Vivado або Quartus, де описуються часові обмеження та проводиться симуляція для підтвердження того, що програмна логіка коректно усуває високочастотні шуми, зберігаючи цілісність структури кадру.

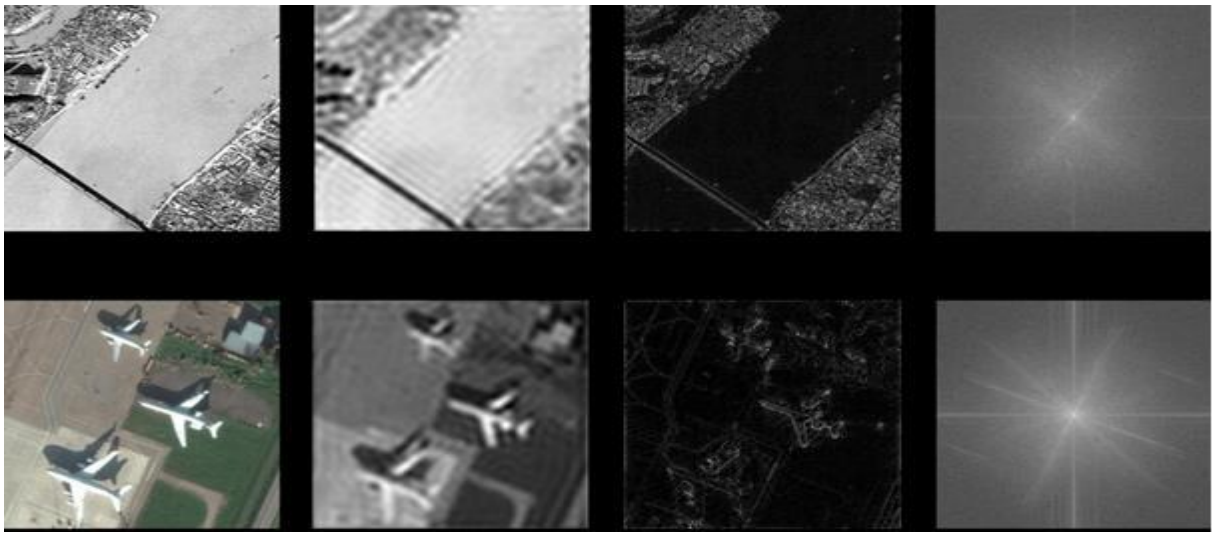


Рисунок 1 - приклади низькочастотних зображень

На відміну від програмної реалізації, HDL використовує потокову обробку, де зображення розглядається як потік цифрових значень. Кожен піксель передається як цифрове слово і обробляється синхронно. Для реалізації просторових фільтрів використовується архітектура ковзного вікна розмірністю 3×3. Навколо поточного пікселя формується область, значення якої

використовуються для обчислення нового значення центрального пікселя за допомогою буферів рядків (line buffers).

Розробка моделі виконувалася у середовищі MATLAB за допомогою мови VHDL. Архітектура фільтра базується на методі просторової фільтрації з коефіцієнтами 1/9 для кожного елемента матриці. Оскільки апаратні засоби мають обмежені ресурси для чисел із рухомою комою, використано арифметику з фіксованою комою (Fixed-Point). Коефіцієнт 1/9 масштабується шляхом цілочисельного множення на константу 14564.

Конвеєрна архітектура (pipeline) дозволяє розбити алгоритм на етапи: формування вектора RGB, множення на матрицю фільтра та акумуляція результату. Для узгодження часу обробки впроваджено каскад регістрів затримки координат (x_out, y_out), що забезпечує синхронізацію вихідних даних із вхідним потоком пікселів [4].

Таблиця 1 - Порівняння підходів до низькочастотної фільтрації

Характеристика	Програмна реалізація (ЦП)	HDL-модель (ПЛІС/FPGA)
Метод обробки	Послідовне виконання інструкцій	Паралельна потокова обробка (Streaming)
Режим часу	Залежить завантаження від ОС	Реальний час (детермінована затримка)
Арифметика	Рухома кома (Floating-Point)	Фіксована кома (Fixed-Point)
Робота з пам'яттю	Прямий доступ за адресою	Буфери рядків (Line Buffers) та регістри зсуву
Обчислення	Стандартна операція ділення	Множення на цілочисельний коефіцієнт
Організація обробки	Цикли та переривання	Конвеєрна архітектура (Pipelining)
Масштабованість	Обмежена ядрами процесора	Повна паралелізація каналів R, G, B

Висновки. Розроблена HDL-модель забезпечує ефективне згладжування цифрових зображень у реальному часі. Використання конвеєризації та паралельної обробки каналів R, G та B дозволяє досягти високої пропускної здатності системи. Застосування мови VHDL забезпечує можливість подальшого синтезу моделі для реалізації на базі ПЛІС (FPGA).

Список літератури

1. Weste N., Harris D. CMOS VLSI Design: A Circuits and Systems Perspective. 4th ed. Boston: Addison-Wesley, 2011. 1024 p.

2.Pedroni V. A. Circuit Design with VHDL. Cambridge: MIT Press, 2010. 640 p.

3.Gonzalez R. C., Woods R. E. Digital Image Processing. 4th ed. Pearson, 2018. 1168 p.

4.Bailey D. G. Design for Embedded Image Processing on FPGAs. Singapore: Wiley, 2011. 488 p.

Білокур В.В.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

ПРОЄКТУВАННЯ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ ВИЯВЛЕННЯ ВТОРГНЕНЬ

Вступ. Сучасні комп'ютерні мережі функціонують в умовах зростання інтенсивності та різноманітності кіберзагроз, зокрема атак на прикладні служби, промислові сегменти, транспортні мережі та канали шифрованого обміну даними. У таких умовах підвищується актуальність систем моніторингу безпеки, здатних не лише фіксувати відомі сигнатури, а й виявляти аномалії та нові сценарії вторгнень на основі аналізу мережевого трафіку. При цьому практичне застосування ускладнюється великими обсягами даних, зміною профілів нормальної поведінки, дисбалансом класів та наявністю зашумлених або неповних ознак, що знижує ефективність традиційних евристичних і статистичних методів.

Перспективним напрямом розв'язання зазначених проблем є використання глибокого навчання та суміжних інтелектуальних підходів, які забезпечують автоматизоване виділення інформативних закономірностей у трафіку та підвищують точність класифікації атак за умови належної підготовки даних. Зокрема, у наукових працях запропоновано моделі на основі рекурентних мереж, механізмів уваги та їхніх гібридних комбінацій для покращення розпізнавання складних часових залежностей у потоках даних [1, 2].

Постановка задачі. Побудова інтелектуальної системи моніторингу мережевої безпеки на основі глибокого аналізу трафіку передбачає формалізацію задачі виявлення вторгнень як задачі класифікації (або детекції аномалій) за сукупністю параметрів мережевих з'єднань. У практичних умовах дані надходять у вигляді потоку подій, тому рішення має забезпечувати обробку як у режимі offline (навчання/валідація на еталонних наборах), так і в режимі online (оперативний моніторинг і фіксація інцидентів). Для відтворюваного оцінювання результатів обрано відкриті набори даних, що широко застосовуються у дослідженнях IDS: CIC-IDS2017 та UNSW-NB15 [3, 4]. Крім того, під час формування та перевірки ознак трафіку доцільним є використання інструментарію аналізу пакетів, зокрема Wireshark [5].

Розглянемо формалізацію задачі. Нехай мережевий трафік представлено як послідовність записів $X = \{x_i\}_{i=1}^{N}$, де кожен запис x_i відповідає потоку або з'єднанню та описується вектором ознак $x_i \in \mathbb{R}^d$. Вектор ознак може включати статистичні, часові та протокольні характеристики (наприклад, кількість пакетів/байтів, тривалість з'єднання, напрямки передачі, прапорці TCP, інтервали між пакетами тощо). Множина класів позначається як $Y = \{0,1,\dots,K\}$, де 0 – нормальний трафік, а $1..K$ – типи атак (за наявності багатокласової розмітки) або узагальнений клас "атака" (у випадку бінарної постановки).

Тоді задача виявлення вторгнень формулюється як побудова функції класифікації

$$f: \mathbb{R}^d \to Y, \tag{1}$$

яка мінімізує помилку класифікації на тестовій вибірці та є стійкою до змін профілю трафіку в часі. Практично це означає, що для кожного нового запису x_t, отриманого в процесі моніторингу, система має обчислити прогноз $\hat{y}_t = f(x_t)$ та рівень впевненості (або ймовірність), після чого ухвалити рішення про наявність інциденту.

Якщо застосовується модель на основі послідовностей (наприклад, BiLSTM/BiGRU), тоді вхід може бути організований як послідовність векторів ознак:

$$s_j = \{x_{j,1}, x_{j,2}, \dots, x_{j,T}\}, \tag{2}$$

де T – довжина вікна спостережень або кількість кроків у часовій агрегації.

Такий підхід узгоджується з моделями, що враховують часові залежності та здатні виділяти інформативні фрагменти через attention [1, 2, 6]. Водночас для системи моніторингу важливо, щоб обрана схема формування послідовностей була однозначною, відтворюваною та придатною до роботи в online-режимі.

Об’єктом дослідження є мережевий трафік комп’ютерних мереж і процеси виникнення та прояву комп’ютерних атак у інформаційно-телекомунікаційних системах.

Предметом дослідження є методи та алгоритми інтелектуального виявлення вторгнень на основі глибокого навчання, а також принципи побудови програмно-алгоритмічної архітектури системи моніторингу безпеки за даними мережевого трафіку.

Мета роботи полягає у проєктуванні нейромережевої моделі виявлення вторгнень для системи моніторингу безпеки комп’ютерної мережі на основі глибокого аналізу трафіку, що забезпечує виявлення вторгнень шляхом навчання та застосування моделей глибокого навчання, а також підтримує збір, підготовку та візуалізацію результатів у процесі моніторингу.

Основний матеріал. Проєктування нейромережевої моделі виявлення вторгнень у межах інтелектуальної системи моніторингу має враховувати дві ключові обставини:

–мережевий трафік характеризується часовою залежністю та контекстом, який не завжди відображається в одиничному записі потоку;

–у практичних умовах важливо забезпечити прийнятний компроміс між точністю детекції та обчислювальними витратами при інференсі.

З огляду на це доцільним є вибір архітектури, здатної моделювати послідовності та акцентувати увагу на найінформативніших фрагментах трафіку.

В якості базового варіант доцільно застосувати гібридну архітектуру BiGRU + MultiHead Attention, оскільки вона поєднує дві важливі властивості (рисунок 1):

– моделювання часових залежностей за рахунок двонапрямної рекурентної мережі (BiGRU/або BiLSTM);

– селективність через механізм уваги, який підсилює внесок найінформативніших кроків послідовності та зменшує вплив шумових фрагментів [1, 2, 6].

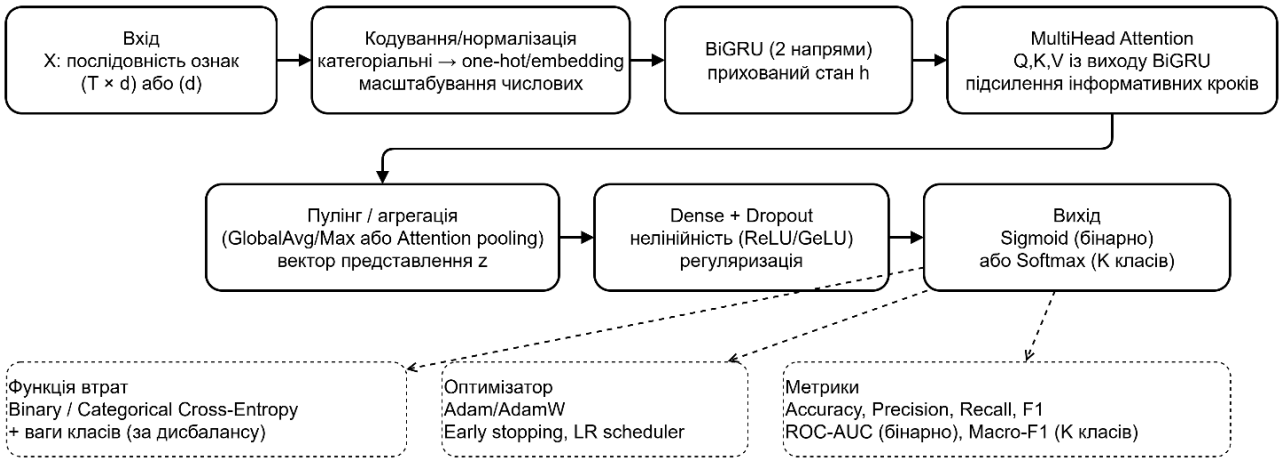


Рисунок 1 – Структурна схема нейромережевої моделі

BiGRU у порівнянні з BiLSTM часто розглядається як більш “економний” варіант з меншою кількістю параметрів при збереженні здатності відстежувати контекст, що є корисним для режиму моніторингу. Використання MultiHead Attention дозволяє моделі одночасно аналізувати кілька “проєкцій” залежностей у послідовності, що підвищує чутливість до різних патернів атак (наприклад, короткі імпульсні прояви та довші поведінкові тренди) [1].

У загальному вигляді модель отримує на вхід або:

– табличний вектор ознак потоку $x \in \mathbb{R}^d$ (спрощена постановка), або

– послідовність $x \in \mathbb{R}^{T \times d}$, сформовану з ковзного вікна/групування потоків (послідовнісна постановка), що є пріоритетною для BiGRU/BiLSTM [2].

Вихід моделі реалізується через Sigmoid (для бінарної детекції «норма/атака») або Softmax (для багатокласової класифікації типів атак), що узгоджується з типовими постановками задач IDS [1, 6].

Для обґрунтування вибору базової архітектури доцільно розглянути альтернативи, які застосовуються в IDS у різних доменах:

1. DBN-подібні підходи (Deep Belief Network). Моделі на основі DBN орієнтовані переважно на табличні ознаки та можуть демонструвати конкурентні результати на класичних наборах даних. У працях, де застосовано DBN у поєднанні з проєкційними методами, підкреслюється потенціал такого підходу для виділення прихованих структур у даних [7]. Водночас DBN зазвичай слабше відображає часовий контекст, якщо послідовність не моделюється явно. Додатковим аргументом на користь часових моделей є результати, де глибока довіра застосовується саме для детекції атак, але потребує ретельного підбору ознак та параметрів [8].

2. Легковагові моделі для доменів із обмеженими ресурсами (MobileViT, CapsNet+SRU). У транспортних мережах (CAN/in-vehicle) важливими є компактність і швидкодія, що обґрунтовує використання MobileViT або інших lightweight-архітектур [9]. Для промислового інтернету (IIoT) запропоновано поєднання CapsNet та SRU, яке орієнтується на виявлення структурних залежностей і роботу з послідовностями [10]. Разом з тим такі архітектури часто є домен-специфічними: модель, оптимізована під CAN або IIoT, може втрачати узагальнювальну здатність при перенесенні в класичний NIDS без адаптації ознак та навчальної вибірки.

3. ELM (Extreme Learning Machine), зокрема несупервізовані варіанти. ELM-підхід привабливий швидким навчанням і відносною простотою, що робить його корисним як базову лінію порівняння [11]. Однак для складних багатокласових сценаріїв і задач із часовими залежностями очікувано кращі результати забезпечують гібридні глибинні моделі, які можуть відображати контекст та нелінійні взаємозв'язки між ознаками [1, 2].

Узагальнюючи, BiGRU + MultiHead Attention є збалансованим рішенням для даної теми: воно враховує часову природу трафіку (на відміну від чисто табличних моделей) і не є жорстко прив'язаним до вузького домену (на відміну від деяких lightweight-рішень), водночас забезпечуючи кращий потенціал точності, ніж спрощені ELM-підходи.

Розглянемо визначення гіперпараметрів, функції втрат, метрик оцінювання.

Функція втрат:

– для бінарної постановки застосовується Binary Cross-Entropy;
– для багатокласової – Categorical Cross-Entropy.

З огляду на типову незбалансованість класів у IDS доцільно передбачити ваги класів у функції втрат, щоб зменшити деградацію recall для міноритарних атак.

В якості стандартного оптимізатора доцільно застосувати Adam/AdamW із планувальником швидкості навчання (LR scheduler) та механізмом ранньої зупинки для обмеження перенавчання.

Метрики оцінювання. Поряд із accuracy доцільно обов'язково аналізувати precision, recall, F1-score, причому у разі багатокласової задачі – macro-F1 (як менш чутливу до дисбалансу). Для бінарної детекції корисним є ROC-AUC, що характеризує якість ранжування рішень при різних порогах.

Також розглянемо механізми адаптивності/оновлення моделі та підвищення продуктивності. У реальному моніторингу модель стикається з дрейфом трафіку (зміна сервісів, поведінки користувачів, появи нових атак). Тому доцільно передбачити контур оновлення, який не порушує стабільності системи:

1. Періодичне перенавчання (наприклад, щотижня/щомісяця) на накопичених верифікованих логах із фіксацією версій моделі та порівнянням метрик. Такий підхід узгоджується з концепціями високопродуктивних адаптивних систем детекції [12].

2. Online/інкрементне оновлення за умов наявності контрольованого механізму валідації, щоб уникнути "зсуву" моделі через шумові або неправильно розмічені дані; це відповідає підходам IDS/IPS із адаптивним навчанням [13].

3. Підвищення продуктивності на рівні моделі: зменшення розмірності ознак (за потреби), обмеження *T*, застосування більш компактних конфігурацій BiGRU, а також оптимізація інференсу (батчування, прискорені бібліотеки). Практично важливо, щоб прискорення не призводило до різкого падіння recall критичних атак.

Висновки. Отже, в даному дослідженні виконано проєктування нейромережевої моделі виявлення вторгнень для інтелектуальної системи моніторингу мережевої безпеки. В якості базової архітектури обґрунтовано використання гібридної моделі BiGRU + MultiHead Attention, яка поєднує здатність рекурентних мереж відображати часовий контекст трафіку та механізм уваги для виділення найбільш інформативних фрагментів послідовності, що є принципово важливим для задач IDS у реальних умовах.

Список літератури

1. Fan J., Ge L., Zhang H., et al. Intrusion Detection Model Integrating MultiHead Attention and BiGRU. Computer and Digital Engineering. 2023. Vol. 51, no. 1. P. 74–80.
2. Yu J. Design of BiLSTM Algorithm for Network Security Intrusion Detection. Microcomputer Applications. 2024. Vol. 40, no. 11. P. 222–225.
3. Canadian Institute for Cybersecurity (CIC). Intrusion detection evaluation dataset (CIC-IDS2017). URL: https://www.unb.ca/cic/datasets/ids-2017.html (дата звернення: 15.02.2026).
4. UNSW. The UNSW-NB15 Dataset (project page). URL: https://research.unsw.edu.au/projects/unsw-nb15-dataset (дата звернення: 15.02.2026).
5. Wireshark Foundation. Wireshark User's Guide (online documentation). URL: https://www.wireshark.org/docs/ (дата звернення: 15.02.2026).
6. Zhao Y., Hu Y. Research and Optimization of Network Intrusion Detection Methods Based on Deep Learning. Computer Programming Techniques and Maintenance. 2024. No. 11. P. 161–164.
7. Wu Y., Li W., Chen Y. Intrusion Detection Model Based on Deep Belief Network Fusion and Local Preserving Projection. Computer Applications and Software. 2024. Vol. 41, no. 6. P. 62–71.
8. Скумін Т., Головко В., Саченко А., Комар М. Застосування нейронних мереж глибокої довіри для виявлення комп'ютерних атак. Захист інформації і безпека інформаційних систем : зб. тез міжнар. наук.-техн. конф. Львів, 2-3 червня, 2016. С. 162-164.
9. Chen H., Zhang L., Jin H., et al. Vehicle CAN Intrusion Detection Method Based on MobileViT Lightweight Network. Information Security Research. 2024. Vol. 10, no. 5. P. 411–420.
10. Li Q., Liu C., Gao G. Industrial Internet Intrusion Detection Method Based on CapsNet and SRU. Computer Technology and Development. 2024. Vol. 34, no. 7. P. 93–99.
11. Fang J., Leng F. Network Security Intrusion Detection System Based on Deep Learning. Procedia Computer Science. 2025. Vol. 261. P. 1107–1113.
12. Komar M., Kochan V., Dubchak L., Sachenko A., Golovko V., Bezobrazov S., Romanets I. High performance adaptive system for cyber attacks detection. The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications : Proceedings (Bucharest, Romania, September 21-23, 2017). Bucharest, 2017. Vol. 2. P. 853-858.
13. Vladov S., Vysotska V., Vashchenko S., Bolvinov S., Glubochenko S., Repchonok A., Korniienko M., Nazarkevych M., Herasymchuk R. Neural Network IDS/IPS Intrusion Detection and Prevention System with Adaptive Online Training to Improve Corporate Network Cybersecurity, Evidence Recording, and Interaction with Law Enforcement Agencies. Big Data and Cognitive Computing. 2025. Vol. 9, no. 11. P. 267.

Вороняк Б.П.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

ПРОГНОЗУВАННЯ ХВОРОБ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Вступ. Сільське господарство є однією з ключових галузей економіки, стабільне функціонування якої безпосередньо впливає на продовольчу безпеку та соціально-економічний розвиток держави. Однією з основних причин зниження врожайності сільськогосподарських культур є поширення різноманітних хвороб рослин, які за відсутності своєчасної діагностики можуть призводити до значних економічних втрат. У зв'язку з цим актуальним завданням є розроблення ефективних методів раннього прогнозування та виявлення захворювань культур.

Традиційні способи діагностики хвороб рослин базуються переважно на візуальному огляді посівів фахівцями або проведенні лабораторних аналізів. Такі підходи, хоча й забезпечують достатню точність, характеризуються високою трудомісткістю, значними часовими витратами та суб'єктивністю оцінювання. Крім того, їх застосування є ускладненим у випадку великих аграрних площ, що обмежує можливість оперативного реагування на появу захворювань.

Розвиток цифрових технологій та методів штучного інтелекту відкрив нові можливості для автоматизації процесів моніторингу стану рослин. Зокрема, методи комп'ютерного зору у поєднанні з глибоким навчанням дозволяють здійснювати аналіз зображень листя та інших частин рослин з метою виявлення характерних ознак захворювань [1-3]. Серед таких методів особливе місце посідають згорткові нейронні мережі, які здатні автоматично формувати інформативні просторові ознаки та демонструють високі показники точності у задачах класифікації зображень [4-6].

Ефективність застосування згорткових нейронних мереж у задачах прогнозування хвороб рослин значною мірою залежить від якості навчальних даних, вибору архітектури моделі та параметрів навчання. Дослідження свідчать, що використання сучасних архітектур глибокого навчання, а також методів аугментації даних і перенесення навчання, дозволяє підвищити узагальнювальну здатність моделей навіть за обмежених обсягів навчальної вибірки [4, 6]. Це робить такі підходи перспективними для практичного використання в аграрних інформаційних системах.

Постановка задачі. Проведений аналіз предметної області та існуючих підходів до прогнозування хвороб сільськогосподарських культур показав, що згорткові нейронні мережі є одним із найбільш ефективних інструментів для автоматизованого аналізу зображень рослин. Сучасні дослідження підтверджують здатність CNN забезпечувати високі показники точності класифікації за рахунок автоматичного виділення просторових ознак, характерних для різних типів захворювань. Водночас більшість наукових робіт зосереджена переважно на вдосконаленні моделей глибокого навчання, тоді як питання практичної реалізації таких підходів у вигляді завершених програмних модулів залишаються недостатньо опрацьованими.

У реальних умовах аграрного виробництва використання методів глибокого навчання потребує не лише наявності ефективної моделі, а й програмного забезпечення, здатного забезпечити повний цикл обробки даних. Такий цикл охоплює завантаження та попередню обробку зображень, виконання прогнозування на основі згорткової нейронної мережі, а також формування результатів у зрозумілому та зручному для користувача вигляді. Недостатня увага до цих аспектів ускладнює впровадження результатів наукових досліджень у практичну діяльність аграрних підприємств.

Об'єктом дослідження є процес автоматизованої діагностики хвороб рослин за цифровими зображеннями.

Предметом дослідження є методи та моделі глибокого навчання, зокрема згорткові нейронні мережі та підходи перенесення навчання, а також програмні засоби їх реалізації.

Метою роботи є проектування програмного модуля прогнозування хвороб сільськогосподарських культур на основі згорткових нейронних мереж, який забезпечує автоматизований аналіз зображень рослин і формування прогнозу щодо їхнього стану з використанням сучасних методів глибокого навчання.

Основний матеріал. Загальна архітектура програмного модуля має багаторівневу структуру та включає кілька логічно відокремлених компонентів, кожен з яких відповідає за виконання окремих функцій. Основними компонентами системи є модуль введення даних, модуль попередньої обробки зображень, модуль прогнозування на основі згорткової нейронної мережі та модуль формування і представлення результатів.

Модуль введення даних забезпечує завантаження зображень рослин у систему та виконує первинну перевірку їх коректності. На цьому етапі здійснюється перевірка формату файлів, розміру зображень та їх доступності для подальшої обробки. Такий підхід дозволяє зменшити ймовірність помилок на наступних етапах роботи програмного модуля.

Модуль попередньої обробки зображень призначений для підготовки вхідних даних до подачі у згорткову нейронну мережу. До його основних функцій належать масштабування зображень до фіксованого розміру, нормалізація значень пікселів, а також виконання операцій, спрямованих на зменшення впливу шумів та варіацій умов зйомки. Реалізація цього модуля є критичною для забезпечення стабільності та узагальнювальної здатності моделі прогнозування [4].

Ключовим компонентом архітектури є модуль прогнозування, який реалізує згорткову нейронну мережу для класифікації стану рослин. Даний модуль відповідає за завантаження попередньо навченої моделі, обробку підготовлених зображень та обчислення ймовірностей належності до кожного з класів. Використання згорткових нейронних мереж у цьому модулі зумовлене їх здатністю ефективно виявляти просторові ознаки, характерні для різних типів захворювань рослин [1-3].

Модуль формування результатів забезпечує інтерпретацію вихідних даних нейронної мережі та їх представлення у зручному для користувача вигляді. Результати прогнозування можуть відображатися у вигляді текстових повідомлень, таблиць або графічних елементів, що містять інформацію про прогнозований клас і відповідні значення ймовірностей. Такий підхід підвищує наочність результатів і полегшує їх подальший аналіз.

Взаємодія між компонентами програмного модуля відбувається послідовно, відповідно до визначеного потоку обробки даних. Зображення, отримані на вході системи, передаються до модуля попередньої обробки, після чого підготовлені дані надходять до модуля прогнозування. Результати обчислень передаються до модуля формування результатів, який завершує цикл роботи системи. Така організація забезпечує прозорість обробки даних та спрощує налагодження й супровід програмного модуля.

Загальна структура програмного модуля прогнозування хвороб сільськогосподарських культур на основі згорткових нейронних мереж може бути представлена у вигляді структурної схеми, що ілюструє основні компоненти системи та напрямки передачі даних між ними.

На рисунку 1 представлено архітектуру програмного модуля прогнозування хвороб сільськогосподарських культур.

Центральним елементом програмного модуля прогнозування хвороб сільськогосподарських культур є модель згорткової нейронної мережі, яка виконує автоматизований аналіз зображень рослин та формує прогноз щодо їхнього стану. Вибір саме згорткової нейронної мережі зумовлений її здатністю ефективно працювати з просторово структурованими даними та виділяти інформативні ознаки, характерні для різних типів захворювань рослин [1-3].

Архітектура розробленої моделі побудована за принципом послідовної обробки даних і складається з кількох функціональних блоків, що включають згорткові шари, шари підвибірки, шар нормалізації, повнозв’язаний шар та вихідний шар класифікації. Загальну структуру згорткової нейронної мережі, використаної у програмному модулі, подано на рисунку 2.

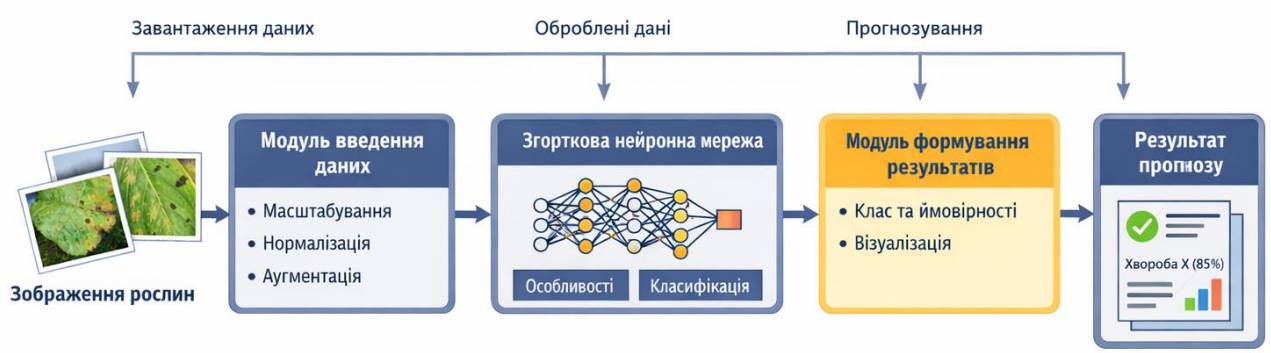


Рисунок 1 – Структурна схема програмного модуля прогнозування хвороб сільськогосподарських культур

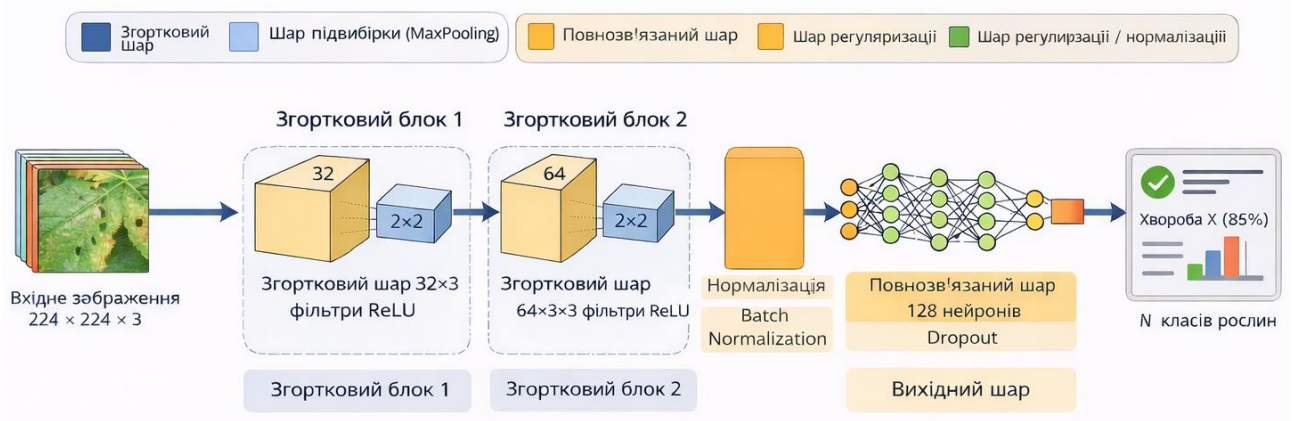


Рисунок 2 – Архітектура згорткової нейронної мережі

На вхід нейронної мережі подаються попередньо оброблені зображення рослин розміром 224×224×3 у форматі RGB. Перший згортковий блок містить згортковий шар із 32 фільтрами розміром 3×3, після якого застосовується функція активації ReLU та шар підвибірки MaxPooling із розміром вікна 2×2. Цей блок призначений для виділення базових просторових ознак, таких як контури та елементарні текстурні елементи.

Другий згортковий блок має аналогічну структуру, проте використовує 64 згорткові фільтри, що дозволяє виділяти складніші та більш абстрактні ознаки, пов’язані з характерними симптомами захворювань рослин. Застосування шару підвибірки на цьому етапі сприяє зменшенню просторової розмірності ознак і зниженню обчислювального навантаження, що є важливим для практичного використання програмного модуля.

Для підвищення стабільності процесу навчання та зменшення внутрішнього зміщення активацій у моделі використовується шар пакетної нормалізації (Batch Normalization). Даний підхід дозволяє пришвидшити збіжність алгоритму навчання та позитивно впливає на узагальнювальну здатність згорткової нейронної мережі [7].

Після завершення згорткових операцій сформовані ознаки передаються до повнозв’язаного шару, який містить 128 нейронів. Цей шар виконує інтеграцію виділених ознак і формує внутрішнє подання, необхідне для виконання класифікації. З метою зменшення ризику перенавчання у повнозв’язаному шарі застосовується регуляризація шляхом випадкового вимкнення нейронів (Dropout) з імовірністю 0,5.

Вихідний шар згорткової нейронної мережі містить N нейронів, де N відповідає кількості класів стану рослин. Для формування кінцевого прогнозу використовується функція активації Softmax, яка дозволяє отримати ймовірнісну оцінку належності зображення до кожного з можливих класів.

Таким чином, розроблена модель згорткової нейронної мережі поєднує відносну простоту архітектури з достатньою виразною здатністю для задач прогнозування хвороб сільськогосподарських культур. Обрана структура забезпечує баланс між точністю

класифікації та обчислювальною ефективністю, що є важливим для подальшої реалізації та експериментального дослідження програмного модуля.

Висновки. Обґрунтовано актуальність розроблення програмного модуля прогнозування хвороб сільськогосподарських культур на основі згорткових нейронних мереж. Визначено, що поширення хвороб рослин є однією з причин зниження врожайності та економічних втрат, а традиційні методи діагностики потребують значних часових витрат, залежать від досвіду фахівців і не завжди придатні для оперативного застосування на великих аграрних площах.

Показано, що використання методів комп'ютерного зору та глибокого навчання дає змогу автоматизувати аналіз зображень рослин і виявляти характерні ознаки захворювань. Особливу увагу приділено згортковим нейронним мережам, які здатні автоматично формувати просторові ознаки зображень і забезпечувати ефективну класифікацію стану рослин.

Сформульовано постановку задачі, визначено об'єкт, предмет і мету дослідження. Обґрунтовано, що для практичного використання методів глибокого навчання необхідно не лише побудувати модель прогнозування, а й спроєктувати програмний модуль, який забезпечує повний цикл обробки даних: завантаження зображення, попередню обробку, прогнозування та подання результату користувачу.

Розглянуто архітектуру програмного модуля, що складається з модуля введення даних, модуля попередньої обробки зображень, модуля прогнозування та модуля формування результатів. Така структура забезпечує послідовну обробку інформації, спрощує налагодження системи та створює умови для подальшого розвитку програмного рішення.

Описано архітектуру згорткової нейронної мережі, яка є центральним компонентом програмного модуля. Визначено основні елементи моделі: згорткові шари, шари підвибірки, шар пакетної нормалізації, повнозв'язаний шар, шар регуляризації та вихідний шар із функцією Softmax. Запропонована структура дозволяє поєднати достатню виразну здатність моделі з прийнятною обчислювальною ефективністю.

Список літератури

1. Ferentinos K. P. Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture. 2018. Vol. 145. Pp. 311–318.
2. Mohanty S. P., Hughes D. P., Salathé M. Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science. 2016. Vol. 7. Article 1419.
3. Ramcharan A., Baranowski K., McCloskey P., Ahmed B., Legg J., Hughes D. P. Deep learning for image-based cassava disease detection. Frontiers in Plant Science. 2017. Vol. 8. Article 1852.
4. Barbedo J. G. A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Computers and Electronics in Agriculture. 2018. Vol. 153. Pp. 46–53.
5. Fuentes A., Yoon S., Kim S. C., Park D. S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. Sensors. 2017. Vol. 17(9). Article 2022. doi:10.3390/s17092022.
6. Too E. C., Yujian L., Njuki S., Yingchun L. A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture. 2019. Vol. 161. Pp. 272–279.
7. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv. 2015. arXiv:1502.03167. doi:10.48550/arXiv.1502.03167.

Омельчук О.А.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

ІНТЕЛЕКТУАЛЬНЕ ВІДЕОСПОСТЕРЕЖЕННЯ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ТА ТУМАННИХ ОБЧИСЛЕНЬ

Вступ. У сучасних умовах стрімкого розвитку цифрових технологій системи відеоспостереження стали важливим складником інформаційної інфраструктури безпеки. Такі системи застосовуються на об'єктах критичної інфраструктури, у транспортних вузлах, громадських просторах, освітніх закладах, промислових підприємствах і приватному секторі. Зростання кількості камер, безперервне надходження великих обсягів відеоданих і потреба в оперативному реагуванні на потенційно небезпечні події зумовили перехід від звичайного пасивного запису відео до інтелектуального автоматизованого аналізу відеопотоку [1, 2].

Традиційні системи відеоспостереження, у яких основний акцент зроблено на передавання та централізоване збереження відео, мають низку суттєвих обмежень. До них належать значне навантаження на мережеву інфраструктуру, висока затримка оброблення, обмежені можливості масштабування, а також залежність від центрального обчислювального вузла. За таких умов знижується ефективність системи в задачах реального часу, де критичне значення мають швидке виявлення об'єктів, подій або аномальних ситуацій та негайне формування повідомлення про них [2-5].

Одним із найбільш перспективних напрямів розв'язання цієї задачі стало поєднання методів глибокого навчання із розподіленими підходами до оброблення даних. Зокрема, згорткові нейронні мережі продемонстрували високу ефективність у задачах комп'ютерного зору: виявленні об'єктів, сегментації зображень, розпізнаванні осіб, аналізі сцен і подій у відеопотоці [6, 7]. Водночас для забезпечення роботи в режимі реального часу важливим стало перенесення частини обчислень ближче до джерела даних, тобто до камер або локальних вузлів опрацювання. Саме тому дедалі ширшого застосування набувають туманні обчислення, які забезпечують проміжний рівень між периферійними пристроями та хмарною інфраструктурою, зменшуючи затримку, обсяг даних, що передаються і навантаження на центральні сервери [4, 8].

Постановка задачі. Проведений аналіз предметної області показав, що сучасні системи відеоспостереження дедалі активніше переходять від простого запису відео до автоматизованого аналізу подій у потоці кадрів. Використання згорткових нейронних мереж суттєво підвищило ефективність виявлення об'єктів, а застосування туманних обчислень створило умови для перенесення частини оброблення ближче до джерела даних. Це особливо важливо для систем, які повинні працювати в режимі реального часу та забезпечувати швидке реагування на події.

Разом із тим аналіз існуючих рішень засвідчив, що більшість із них мають певні обмеження. Частина систем орієнтована переважно на централізоване збереження та перегляд відео, тому не забезпечує достатнього рівня автоматизації аналізу. Інші рішення реалізують інтелектуальне виявлення об'єктів, але потребують значних апаратних ресурсів або не враховують особливості розподіленого оброблення даних. Також поширеними є системи, у яких увагу зосереджено на окремому алгоритмі чи моделі, без побудови цілісної програмної архітектури, придатної до практичного використання в умовах безперервного відеопотоку.

Для задач інтелектуального відеоспостереження реального часу важливими є одразу кілька характеристик: здатність системи своєчасно обробляти потік кадрів, автоматично виявляти об'єкти у сцені, інтерпретувати результат виявлення як подію прикладного рівня та забезпечувати подальше збереження або передавання результату. За відсутності такої узгодженої роботи програмних компонентів система або втрачає оперативність, або зводиться лише до часткового розв'язання задачі.

Об'єктом дослідження є процес інтелектуального оброблення відеопотоку в системах

відеоспостереження реального часу. Предметом дослідження є методи, моделі та програмні засоби аналізу відеоданих на основі згорткових нейронних мереж у середовищі туманних обчислень. Метою роботи є проєктування програмної системи інтелектуального відеоспостереження, що забезпечує аналіз відеопотоку в режимі реального часу на основі згорткових нейронних мереж і туманних обчислень.

Основний матеріал. Узагальнену архітектуру програмної системи спроєктовано у вигляді кількох взаємопов'язаних модулів, кожен із яких виконує чітко визначену функцію (рисунок 1).

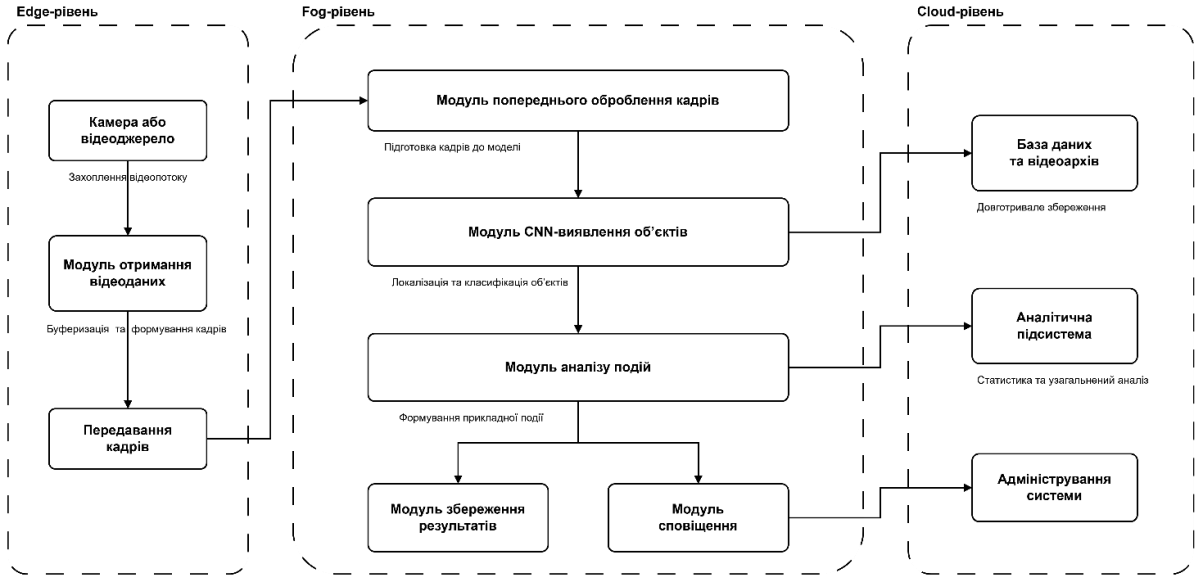


Рисунок 1 – Загальна архітектура програмної системи інтелектуального відеоспостереження

Першим функціональним компонентом є модуль отримання відеоданих. Його призначення полягає у підключенні до джерела відео та формуванні послідовності кадрів для подальшої обробки. Як джерело можуть використовуватися IP-камера, вебкамера або тестовий відеофайл. На цьому етапі система повинна забезпечувати стабільне надходження кадрів, їх зчитування у правильному порядку та передавання до наступного модуля без порушення часової послідовності. Цей модуль є початковою точкою роботи всієї системи, тому від його коректного функціонування залежить стабільність подальшого аналізу.

Другим компонентом є модуль попереднього оброблення кадрів. Він призначений для приведення відеоданих до формату, придатного для аналізу нейронною мережею. У його межах виконуються масштабування кадру, зміна роздільної здатності, перетворення кольорового простору за потреби, нормалізація вхідних даних та усунення надлишкових елементів, які можуть ускладнювати подальше оброблення. Важливість цього модуля полягає в тому, що якість попередньої підготовки кадрів безпосередньо впливає на точність роботи моделі виявлення об'єктів.

Третім і центральним компонентом системи є модуль виявлення об'єктів на основі згорткової нейронної мережі. Саме в ньому реалізується інтелектуальна складова всієї системи. На вхід цього модуля подається підготовлений кадр, а на виході формується набір знайдених об'єктів із координатами, класами та оцінкою впевненості. У контексті цієї роботи основною задачею такого модуля є виявлення цільових об'єктів у кадрі з прийнятною швидкодією. Тому модель повинна бути достатньо точною, але водночас придатною до використання в умовах обмежених обчислювальних ресурсів fog-рівня.

Після отримання результатів детекції вони передаються до модуля аналізу подій. Його функція полягає в інтерпретації результатів нейромережевого виявлення з позиції прикладної логіки. Якщо модуль CNN-виявлення визначає технічний факт наявності об'єкта в кадрі, то модуль аналізу подій визначає змістовний факт події. Наприклад, саме тут перевіряється, чи належить знайдений об'єкт до контрольованого класу, чи виконуються умови фіксації події, чи

слід передати інформацію на збереження та чи потрібно згенерувати повідомлення. Таким чином, цей модуль є сполучною ланкою між комп'ютерним зором і прикладною логікою системи спостереження.

П'ятим компонентом є модуль збереження результатів. Його призначення полягає у фіксації результатів роботи системи у структурованому вигляді. До таких результатів можуть належати кадри з виявленими об'єктами, службові метадані, записи журналу подій, часові мітки, класи об'єктів та рівні впевненості детекції. Збереження результатів є необхідним не лише для архівування, а й для подальшого аналізу роботи системи, перевірки коректності її функціонування та формування статистичних звітів.

Шостим компонентом є модуль сповіщення, який відповідає за формування реакції системи на виявлену подію. Його робота може полягати у відображенні повідомлення в інтерфейсі, створенні текстового запису, передаванні сигналу до зовнішньої підсистеми або в іншій формі інформування користувача.

Ключовим елементом програмної системи інтелектуального відеоспостереження є модель оброблення відеопотоку, яка забезпечує перетворення вхідного кадру у результат сегментації та подальше виділення цільового об'єкта. На відміну від спрощеного підходу, за якого результатом роботи моделі є лише факт наявності об'єкта або його прямокутна область, у цій роботі використано модель, орієнтовану на побудову маски об'єкта. Такий підхід є доцільним для задач інтелектуального відеоспостереження, оскільки дозволяє точніше відокремити об'єкт від фону та підвищити інформативність подальшого аналізу.

Для розроблюваної системи обрано модель на основі згорткової нейронної мережі типу encoder–decoder. Вона забезпечує покадрове оброблення відеопотоку, у межах якого кожен кадр проходить послідовні етапи стискання ознак, формування компактного представлення сцени, відновлення просторової структури та побудови вихідної маски. Після цього за допомогою окремого блоку Mask-Crop формується виділена область об'єкта, яка може бути використана для подальшого прикладного аналізу. Структуру розробленої моделі подано на рисунку 2.

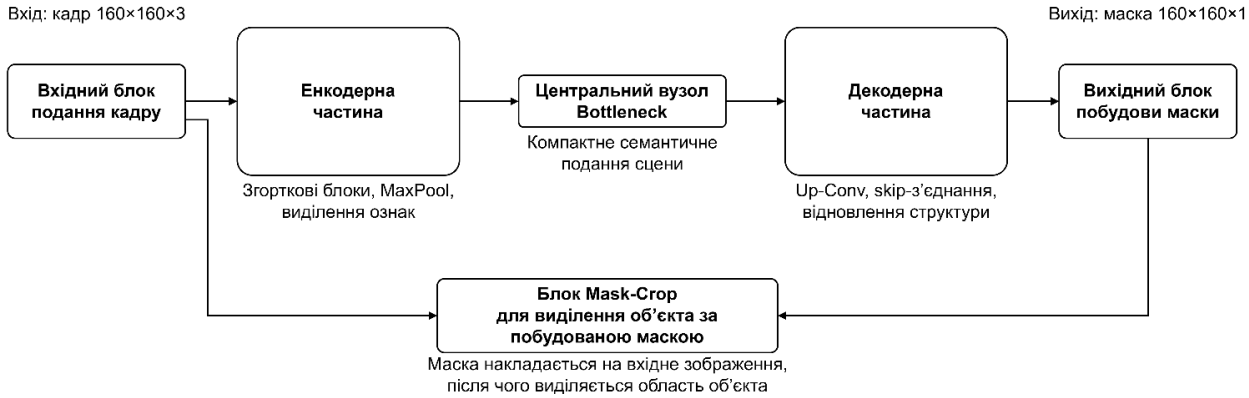


Рисунок 2 – Структура моделі оброблення відеопотоку на основі згорткової нейронної мережі encoder–decoder

Вхідний блок подання кадру призначений для приймання поточного зображення з відеопотоку та приведення його до формату, сумісного з нейромережевою моделлю. У межах цієї роботи кадр розглядається як базова одиниця аналізу. Перед поданням у мережу він має бути нормалізований і приведений до фіксованого розміру. Таким чином, вхідний блок формує уніфіковане подання даних, на основі якого працюють усі наступні компоненти моделі.

Енкодерна частина виконує послідовне виділення ознак із вхідного кадру. На цьому етапі мережа переходить від початкового зображення до багаторівневого ознакового подання сцени. У процесі такого переходу просторовий розмір карт ознак зменшується, а кількість семантичної інформації збільшується. Саме енкодер забезпечує формування високорівневих ознак, що відображають форму, контури та розміщення об'єкта в кадрі. У практичному сенсі цей блок виконує стискання вхідної інформації з одночасним посиленням її змістовності.

Центральний вузол bottleneck є найглибшим рівнем моделі. У ньому формується компактне семантичне представлення сцени, яке містить найбільш узагальнену інформацію про об’єкт і фон. Цей блок виконує роль проміжної ланки між енкодером і декодером. Його призначення полягає не лише в подальшому стисканні ознак, а й у збереженні найбільш значущої інформації, необхідної для подальшого відновлення просторової структури об’єкта.

Декодерна частина виконує зворотне перетворення ознак. Якщо енкодер поступово зменшував просторову роздільну здатність, то декодер відновлює її, поєднуючи глибокі семантичні ознаки з інформацією про локальну структуру об’єкта. Саме на цьому етапі формується наближене просторове представлення маски. Декодер є критично важливим для сегментації, оскільки він забезпечує повернення від стислого ознакового подання до зображення, у якому можна визначити межі об’єкта по пікселях.

Вихідний блок побудови маски формує результат роботи моделі у вигляді одноканальної маски. У такій масці одна частина пікселів відповідає об’єкту або області дії, а інша — фону. Саме цей блок завершує нейромережевий цикл оброблення кадру. На відміну від класичної детекції, де результатом є координати рамки, у цьому випадку результатом є просторово точніше піксельне подання об’єкта.

Після побудови маски використовується блок Mask-Crop, який виконує виділення об’єкта за сформованою маскою. Його функція полягає у накладанні маски на початковий кадр і відокремленні цільової області від надлишкового фону. Унаслідок цього формується очищене представлення об’єкта, яке є більш придатним для наступного етапу аналізу, ніж повний кадр. Такий підхід зменшує вплив фонового шуму та підвищує інформативність оброблення.

Розроблена модель має послідовну логіку функціонування. Спочатку поточний кадр подається на вхідний блок, потім проходить через енкодер і bottleneck, після чого відновлюється в декодері та завершується побудовою маски. Далі блок Mask-Crop формує виділений фрагмент об’єкта. Таким чином, результатом роботи моделі є не лише ознака наявності цільового об’єкта, а конкретна сегментована область, яка може бути використана в системі для формування події або збереження результату.

Висновки. Спроєктовано модульну архітектуру системи, яка охоплює отримання відеоданих, попереднє оброблення кадрів, нейромережевий аналіз, формування подій, збереження результатів і сповіщення. Запропоновано модель оброблення відеопотоку на основі CNN типу encoder–decoder, яка забезпечує побудову маски об’єкта та його виділення за допомогою блоку Mask-Crop. Визначено, що використання туманних обчислень дає змогу перенести основні операції аналізу ближче до джерела відеоданих, зменшити затримку оброблення та підвищити оперативність реагування системи.

Список літератури

1. Chen N., Chen Y., You Y., Ling H., Liang P., Zimmermann R. Dynamic Urban Surveillance Video Stream Processing Using Fog Computing. Proceedings of the IEEE Second International Conference on Multimedia Big Data (BigMM). 2016. P. 105–112.

2. Nasir M., Muhammad K., Lloret J., Sangaiah A. K., Sajjad M. Fog Computing Enabled Cost-Effective Distributed Summarization of Surveillance Videos for Smart Cities. Journal of Parallel and Distributed Computing. 2019. Vol. 126. P. 161–170.

3. Abbas N., Zhang Y., Taherkordi A., Skeie T. Mobile Edge Computing: A Survey. IEEE Internet of Things Journal. 2018. Vol. 5, no. 1. P. 450–465.

4. Chiang M., Ha S., Risso F., Zhang T., Chih-Lin I. Clarifying Fog Computing and Networking: 10 Questions and Answers. IEEE Communications Magazine. 2017. Vol. 55. P. 18–20.

5. De Donno M., Tange K., Dragoni N. Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog. IEEE Access. 2019. Vol. 7. P. 150936–150948.

6. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. P. 770–778.

7. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. P. 779–788.

8. Li C., Xue Y., Wang J., Zhang W., Li T. Edge-Oriented Computing Paradigms: A Survey on Architecture Design and System Management. ACM Computing Surveys. 2018. Vol. 51. Art. 39.

Литвин А. А.
аспірант кафедри АСУ, ІКНІ НУ “Львівська політехніка”
Науковий керівник д.т.н., проф. Цмоць І. Г., кафедра АСУ НУ “Львівська політехніка”

МЕТОДИ ТА ЗАСОБИ ПІДВИЩЕННЯ ШВИДКОДІЇ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ СИСТЕМ КОМП’ЮТЕРНОГО ЗОРУ

Вступ. Згорткові нейронні мережі є основним інструментом сучасних систем комп'ютерного зору - від медичної діагностики до автономних транспортних засобів. Проте висока обчислювальна складність згорткових нейронних мереж обмежує їх застосування в системах реального часу: в типових архітектурах операції згортки та пулінгу займають понад 80–90% загального часу обчислень, що зумовлює активний пошук ефективних методів апаратного прискорення на платформах FPGA. Платформи FPGA вирізняються енергоефективністю, рекурентністю та широкими можливостями паралельної обробки, що робить їх перспективними для прискорення згорткових нейронних мереж [1].

Постановка задачі. Об'єктом дослідження є базові операції згорткових нейронних мереж - двовимірна згортка та пулінг. Предметом дослідження є методи та апаратні засоби прискореного обчислення цих операцій для систем комп'ютерного зору. Метою роботи є розроблення апаратно-орієнтованих методів і структур апаратних засобів для підвищення швидкодії згорткових нейронних мереж, придатних до реалізації на FPGA та GPU зі збереженням необхідної точності обчислень.

Основний матеріал. Для прискорення обчислення двовимірної згортки запропоновано паралельний таблично-алгоритмічний метод. Метод зводить двовимірну згортку до скалярного добутку шляхом розгортання ядра фільтра розміром $n \times m$ та вікна вхідного зображення у вектори довжиною $L = n \cdot m$. Замість L операцій множення використовуються попередньо обчислені таблиці мантис макрочасткових добутків, що базуються на попередньому формуванні часткових добутків із аналізу розрядних зрізів вхідних операндів [2].

Обчислення мантиси скалярного добутку виконується за формулою: $M(\mathrm{Y}) = 2^{-(h-1)} \cdot \sum P_{Ms} \cdot 2^{s}$, де h - розрядність пікселів, PMs - макрочастковий добуток для s-го розрядного зрізу вхідного вектора. Порядок результату визначається як $E(\mathrm{Y}) = E_{\max} + v$, де $E_{\max}$ - найбільший порядок вагових коефіцієнтів, v - кількість розрядів переповнення.

З метою зменшення обсягу пам'яті таблиць вектор довжиною L розбивається на d груп меншої довжини. Коефіцієнт зменшення пам'яті: $K(d) = 2^{L} / (d \cdot 2^{L/d})$. Чисельне моделювання для ядра 4×4 ($L = 16$) показало, що при $d = 2$ досягається зменшення обсягу пам'яті у 128 разів при збільшенні часу обчислень лише на 30%. Дослідження точності підтвердило, що розбиття на групи практично не впливає на похибку: значення MAE залишається стабільним для всіх допустимих значень d, - тобто зменшення апаратних витрат не погіршує точність результату.

Для прискорення операцій пулінгу розроблено метод вертикально-паралельного обчислення максимального та глобального пулінгу. Метод базується на паралельному порівнянні елементів стовпців двовимірного масиву за допомогою дерева компараторів, що скорочує час обчислення з $O(n \cdot m)$ до $O(\log_2(n))$. Метод усередненого пулінгу реалізує накопичувальне підсумовування елементів вікна з діленням на константу, яке при розмірах вікна кратних степеню двійки замінюється зсувом вправо. Усі розроблені методи пулінгу масштабуються нарощуванням рівнів дерева та підтримують конвеєрну обробку - один результат за такт після заповнення конвеєра [3].

Розроблені методи реалізуються як спеціалізовані апаратні пристрої на FPGA: таблично-алгоритмічний метод - як конвеєрна структура з блоками пам'яті типу блоки пам'яті FPGA та деревом суматорів; пулінг - як паралельне дерево компараторів або суматорів залежно від типу операції. Табличний підхід дозволяє легко адаптуватись до різних розмірів ядра шляхом зміни глибини таблиць або кількості груп d, а незалежність таблиць груп забезпечує їх паралельне зчитування.

Оцінка апаратних витрат показує, що запропонований підхід знижує кількість задіяних апаратних множників у L разів порівняно з класичною реалізацією на основі апаратного множення. Використання логічних ресурсів FPGA замість спеціалізованих апаратних множників дозволяє досягти до 78% скорочення їх споживання при збереженні прийнятної точності [4].

Програмна реалізація розроблених методів виконана мовою Python. Розроблено: програму обчислення таблиці мантис макрочасткових добутків; імітаційну модель залежності обсягу пам'яті та часу формування від числа груп d; програму чисельного моделювання з побудовою графіків залежностей. Порівняння класичного та таблично-алгоритмічного методів підтвердило адекватність запропонованого підходу та узгодженість аналітичної моделі з емпіричними результатами.

Дослідження впливу розрядності пікселів h та формату вагових коефіцієнтів на точність показало, що застосування формату float32 забезпечує прийнятну похибку обчислень при мінімальних апаратних витратах. Збільшення розрядності h з 4 до 16 біт суттєво зменшує середню абсолютну похибку MAE, тоді як кількість груп d практично не впливає на точність — це підтверджує, що параметр d можна вільно обирати виключно з міркувань оптимізації апаратних ресурсів без втрати якості результату.

На відміну від реалізацій на GPU, які потребують значних обчислювальних та енергетичних ресурсів, FPGA-реалізація розроблених методів забезпечує детерміновану затримку обробки та низьке енергоспоживання, що є критичними вимогами для вбудованих систем комп'ютерного зору. Крім того, реконфігурованість FPGA дозволяє адаптувати апаратну архітектуру до різних конфігурацій згорткових нейронних мереж без зміни самої схеми пристрою, що суттєво скорочує час виведення продукту на ринок порівняно з розробкою спеціалізованих інтегральних схем (ASIC).

Висновки. Наукова новизна роботи полягає у розробленні: паралельного таблично-алгоритмічного методу обчислення двовимірної згортки з параметром групування d, що забезпечує масштабований компроміс між обсягом пам'яті та часом обчислень; методу вертикально-паралельного пошуку максимуму для операцій пулінгу; методу конвеєрного усередненого пулінгу із заміною ділення зсувом. Практичне значення полягає у можливості апаратної реалізації базових операцій згорткових нейронних мереж на FPGA з підвищеною у L разів швидкодією та зменшеними апаратними витратами, що робить розроблені засоби придатними для систем комп'ютерного зору реального часу.

Список літератури

1. Gao L., Luo Z., Wang L. Convolutional Neural Network Acceleration Techniques Based on FPGA Platforms: Principles, Methods, and Challenges // Information. - 2025. - Vol. 16, No. 10. - P. 914. https://doi.org/10.3390/info16100914

2. Tsmots I., Teslyuk V., Kryvinska N. et al. Development of a generalized model for parallel-streaming neural element and structures for scalar product calculation devices // The Journal of Supercomputing. -2023. - Vol. 79, No. 5. - P. 4820–4846. https://doi.org/10.1007/s11227-022-04838-0

3. Jiang J., Zhou Y., Gong Y., Yuan H., Liu S. FPGA-based Acceleration for Convolutional Neural Networks: A Comprehensive Review // arXiv preprint. - 2025. - arXiv:2505.13461. https://arxiv.org/abs/2505.13461

4. Kim Y. FPGA-Based Convolutional Neural Network Accelerator with Resource-Optimized Approximate Multiply-Accumulate Unit // Electronics. - 2021. - Vol. 10, No. 22. - P. 2859. https://doi.org/10.3390/electronics10222859

Лихтей В.В.(1), Андрійчук Д.Р.(2)
(1)студент 4 курсу ФКІТ ЗУНУ, (2)студент 4 курсу ФКІТ ЗУНУ
Науковий керівник викладач Дериш Б.Б., кафедра КІ ЗУНУ

ІНТЕЛЕКТУАЛЬНА СИСТЕМА РОЗПІЗНАВАННЯ ЗАГРОЗ У СЕРЕДОВИЩІ «РОЗУМНОГО БУДИНКУ»

Вступ. Стрімкий розвиток технологій Інтернету речей та автоматизованих систем керування призвів до широкого впровадження концепції «розумного будинку» у побутову сферу. Сучасні Smart Home системи забезпечують автоматизацію освітлення, клімат-контролю, відеоспостереження та управління доступом до приміщень. Разом із розширенням функціональних можливостей зростає кількість потенційних загроз, пов'язаних із несанкціонованим проникненням, пожежонебезпечними ситуаціями, витоком газу та кіберфізичними атаками на IoT-пристрої [1]. Традиційні системи сигналізації використовують переважно фіксовані алгоритми реагування, що не дозволяє ефективно адаптуватися до змін умов функціонування середовища. У зв'язку з цим актуальним є створення інтелектуальних систем розпізнавання загроз, здатних аналізувати дані сенсорів у режимі реального часу та автоматично визначати небезпечні події.

Постановка задачі. Об'єктом дослідження є процеси моніторингу та виявлення загроз у системах «розумного будинку». Предметом дослідження виступають програмно-апаратні методи аналізу даних сенсорної мережі та алгоритми інтелектуального виявлення небезпечних подій. Метою роботи є розробка системи розпізнавання загроз у середовищі «розумного будинку» на основі аналізу даних IoT-сенсорів та методів машинного навчання для підвищення ефективності функціонування системи безпеки.

Основний матеріал. Запропонована система базується на використанні комплексу IoT-сенсорів, що здійснюють моніторинг параметрів безпеки житлового приміщення. До складу системи входять датчики руху PIR, сенсори диму MQ-2, температурні модулі DHT22 та модулі контролю доступу. Передача інформації між пристроями реалізована із використанням протоколу MQTT, що забезпечує стабільний обмін даними між компонентами системи [2].

Центральний модуль системи виконує збір, обробку та аналіз даних від сенсорів. Для виявлення потенційних загроз використовується алгоритм класифікації подій, який аналізує поточні параметри середовища та порівнює їх із допустимими пороговими значеннями. Формально модель оцінювання рівня небезпеки можна представити у вигляді:

$$R = \sum_{i=1}^{n} w_i \cdot s_i,$$

де w_i – ваговий коефіцієнт відповідного сенсора, s_i – рівень спрацювання датчика.

Для підвищення точності виявлення небезпечних ситуацій застосовано елементи машинного навчання, зокрема алгоритм Random Forest, який дозволяє класифікувати події за ознаками сенсорних даних. Процес класифікації можна подати у вигляді:

$$Y = f(X),$$

де X – вектор параметрів сенсорної системи, Y – результат класифікації події.

У випадку виявлення загрози система автоматично формує повідомлення тривоги та передає його користувачу через вебінтерфейс або мобільний застосунок. Додатково реалізовано механізм журналювання подій для подальшого аналізу інцидентів та оцінювання ефективності роботи системи. Застосування методів інтелектуального аналізу даних дозволяє знизити кількість помилкових спрацювань та підвищити достовірність розпізнавання небезпечних ситуацій [3].

Висновки. У результаті проведеного дослідження розроблено інтелектуальну систему розпізнавання загроз у середовищі «розумного будинку», яка забезпечує комплексний моніторинг параметрів безпеки та автоматизоване виявлення небезпечних подій у режимі реального часу. Практична цінність роботи полягає у можливості інтеграції запропонованого рішення у сучасні IoT-платформи та підвищенні рівня захисту житлових приміщень. Використання алгоритмів машинного навчання дозволяє покращити точність класифікації загроз та забезпечити оперативне реагування на потенційно небезпечні ситуації.

Список літератури

1. S. Siboni, Y. Shabtai, “Security and Privacy in Smart Homes: Challenges and Solutions”, IEEE Access, 2023.

2. A. Alqahtani, M. Alrashidi, “IoT-Based Smart Home Monitoring Systems: Recent Advances and Security Issues”, Sensors, 2022.

3. Y. Ren, W. Sun, “Machine Learning-Based Intrusion Detection for Smart Home Security”, IEEE Access, 2023.

Галунька Б.В.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й. кафедра КІ ЗУНУ

КЛАСИФІКАЦІЯ ЕМОЦІЙ ЗА ВИРАЗОМ ОБЛИЧЧЯ: АРХІТЕКТУРНІ ПІДХОДИ, НАБОРИ ДАНИХ ТА ОЦІНКА ЕФЕКТИВНОСТІ

Вступ. У сучасному технологічному світі автоматизоване розпізнавання емоцій людини за виразом обличчя стає невід'ємною складовою інтелектуальних систем людино-машинної взаємодії. Стрімкий розвиток глибокого машинного навчання відкрив нові можливості для точної та ефективної класифікації емоційних станів безпосередньо на пристроях кінцевого користувача. У даній роботі розглядаються ключові архітектурні підходи до класифікації емоцій, проблеми, пов'язані з дисбалансом навчальних наборів даних, а також методи підвищення ефективності компактних нейромережевих моделей.

Постановка задачі. Об’єкт дослідження - процес автоматичної класифікації емоційних станів людини за зображеннями обличчя. Предмет дослідження - архітектурні підходи до побудови компактних нейромережевих моделей класифікації емоцій та методи підвищення їх ефективності в умовах дисбалансу навчальних даних. Мета роботи - дослідити та порівняти сучасні підходи до побудови програмних модулів класифікації емоцій за зображеннями обличчя, зокрема з використанням оптимізованих згорткових нейронних мереж, а також обґрунтувати вибір архітектурних рішень для роботи в режимі реального часу на пристроях з обмеженими обчислювальними ресурсами.

Основний матеріал.

Задача автоматичного розпізнавання емоцій за виразом обличчя традиційно зводиться до задачі багатокласової класифікації зображень. Теоретичним фундаментом є дискретна модель базових емоцій П. Екмана, яка виокремлює сім категорій: радість, сум, гнів, страх, здивування, огида та нейтральний стан. Для програмних систем нейтральний стан є важливим класом-еталоном для калібрування алгоритмів. Анатомічну формалізацію мімічних виразів забезпечує система FACS (Facial Action Coding System), де кожен рух декомпозується на мінімальні одиниці - рухові одиниці (Action Units, AU), що відповідають скороченням конкретних м'язових груп.

Порівняльний аналіз технологій виявляє три основні підходи до вирішення задачі. Перший - класичні алгоритми машинного навчання: метод опорних векторів (SVM) та алгоритм випадкового лісу (Random Forest) у поєднанні з ручними дескрипторами (HOG, LBP). Ці методи характеризуються мінімальними обчислювальними вимогами, проте демонструють незадовільну точність (70–80 %) за умов відхилення від ідеальних умов зйомки. Другий підхід - хмарні API-сервіси (Amazon Rekognition, Microsoft Azure Face API) - забезпечують дуже

високу точність (до 95 %), але мають критичну залежність від мережевого з'єднання та несуть ризик порушення конфіденційності біометричних даних. Третій підхід - локальні згорткові нейронні мережі (Edge AI) - є оптимальним компромісом між точністю, швидкодією та автономністю.

Серед архітектур глибокого навчання для задач класифікації емоцій виділяються такі: VGG-16 з 138 мільйонами параметрів (точність ~73 % на FER2013), ResNet-50 із залишковими з'єднаннями (73–76 %), MobileNet з технологією глибинно-сепарабельної згортки, що зменшує обчислювальну складність у 8–9 разів, та Mini-Xception - спеціалізована компактна архітектура (близько 58 000 параметрів), розроблена О. Аріагою для роботи в режимі реального часу на пристроях з обмеженими ресурсами. Глибинно-сепарабельна згортка декомпозує стандартну операцію на глибинну (один фільтр - один канал) та точкову (1×1), що уможливлює роботу у режимі реального часу на звичайному процесорі.

Критичною проблемою практичної реалізації є дисбаланс навчальних наборів даних. Зокрема, у широко використовуваному датасеті FER2013 (35 887 зображень 48×48 пікселів) клас «радість» містить ~7 200 прикладів, тоді як «огида» - лише ~440. Для подолання цього дисбалансу застосовуються: аугментація даних (повороти, дзеркальне відображення, зміна яскравості), функція Focal Loss із модулюючим множником $(1 - p_t)^\gamma$, та метод MixUp, який формує синтетичні лінійні комбінації пар зображень разом з відповідними мітками. Для підвищення здатності моделі динамічно зосереджуватись на інформативних ознаках застосовується блок каналової уваги Squeeze-and-Excitation (SE), що за рахунок лише близько 2 000 додаткових параметрів дозволяє отримати помітне приросте точності.

Дослідження за методологією ablation study виявило, що конфігурація Mini-Xception з інтегрованим SE-блоком досягає точності 63,37 % на тестовому наборі FER2013 при 60 312 параметрах - найкращий показник серед легких конфігурацій. Для порівняння, трансферне навчання на основі ResNet-50 з вагами ImageNet дало нижчий результат (59,57 %), оскільки семантичний розрив між загальними зображеннями ImageNet та специфічними мімічними виразами FER2013 виявився занадто значним.

Висновки. Автоматизована класифікація емоцій за виразом обличчя є перспективним та практично затребуваним напрямком, що знаходить застосування у системах моніторингу водіїв, дистанційному навчанні, медичних інформаційних системах та маркетинговій аналітиці. Порівняльний аналіз підходів засвідчує, що локальні оптимізовані згорткові нейронні мережі - зокрема, архітектура Mini-Xception з блоком каналової уваги Squeeze-and-Excitation - забезпечують оптимальний баланс між точністю (63,37 % на FER2013), компактністю (60 000 параметрів) та здатністю до виконання в режимі реального часу на CPU-обладнанні без залежності від мережевої інфраструктури. Запропонована модифікована архітектура підтверджує доцільність цілеспрямованих компактних рішень над трансферним навчанням для задач зі специфічним характером вхідних даних.

Список літератури

1. Li S., Deng W. Deep facial expression recognition: A survey. IEEE Transactions on Affective Computing. 2022. Vol. 13, No. 3. P. 1195–1215.

2. Kollias D., Zafeiriou S. Affect analysis in-the-wild: Valence-arousal, expressions, action units and a unified framework. arXiv:2103.15792. 2021.

3. Xue F., Wang Q., Tan Z. et al. Transfer learning for facial expression recognition via parameter debating and weighting. Pattern Recognition. 2022. Vol. 131. Article 108861.

4. Savchenko A. V. Facial expression- and skin color-based human emotion recognition with class-specific residual networks. IET Biometrics. 2022. Vol. 11, No. 4. P. 337–347.

5. Zhang H., Avrithis Y., Furon T., Amsaleg L. Walking on minimax paths for k-nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2023. Vol. 45, No. 7. P. 8779–8793.

Ковальський А.А., Вишневський Ю.Ю.
студенти 6 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Маслияк Ю.Б., кафедра КН ЗУНУ

АРХІТЕКТУРА ПРОГРАМНОГО МОДУЛЯ ДЛЯ ОПТИМІЗАЦІЇ УПРАВЛІННЯ ЗАПАСАМИ КАБЕЛЬНОЇ ПРОДУКЦІЇ НА ОСНОВІ ГЕНЕТИЧНИХ АЛГОРИТМІВ

Вступ. Стрімкий розвиток сучасних логістичних систем та платформ управління складом (WMS) вимагає впровадження інтелектуальних методів управління матеріальними запасами. В умовах обробки великих масивів замовлень на нарізку кабельно-провідникової продукції застосування традиційних «жадібних» алгоритмів (наприклад, First Fit) призводить до нераціонального розрізання нових заводських барабанів та накопичення значних обсягів неліквідних технологічних відходів [1]. Завдяки використанню методів комбінаторної оптимізації, зокрема еволюційних обчислень, можна досягти суттєвого зниження матеріальних втрат та підвищення операційної ефективності складу.

Постановка задачі. Об’єкт дослідження – процес управління складськими запасами та одновимірного розкрою кабельно-провідникової продукції на логістичних підприємствах. Предмет дослідження – архітектура програмного модуля та алгоритмічні моделі еволюційного пошуку для оптимізації розкрою залишків кабелю. Головна мета даного дослідження полягає в проектуванні та програмній реалізації багаторівневої архітектури автономного мікросервісу на основі генетичних алгоритмів, що забезпечує мінімізацію технологічних відходів та безшовну інтеграцію з існуючими корпоративними ERP/WMS системами.

Основний матеріал. Проектування архітектури програмного модуля оптимізації управління запасами кабельної продукції є важливим етапом конструювання програмного забезпечення. Оскільки розроблюваний модуль має функціонувати в реальних умовах складського комплексу, його архітектура повинна забезпечувати високу продуктивність, надійність, гнучкість до змін бізнес-логіки, а також простоту інтеграції з існуючими корпоративними системами управління підприємством (системами управління складом та планування ресурсів підприємства).

Для забезпечення модульності та дотримання стандартів об'єктно-орієнтованого конструювання програмного забезпечення, внутрішня структура коду модуля розділена на логічні простори імен (пакети). Це дозволяє уникнути сильної зв'язності між компонентами та спрощує командну розробку і подальший супровід продукту [2]. На рис. 1 наведено UML-діаграму пакетів, яка відображає ієрархію просторів імен розробленого рішення.

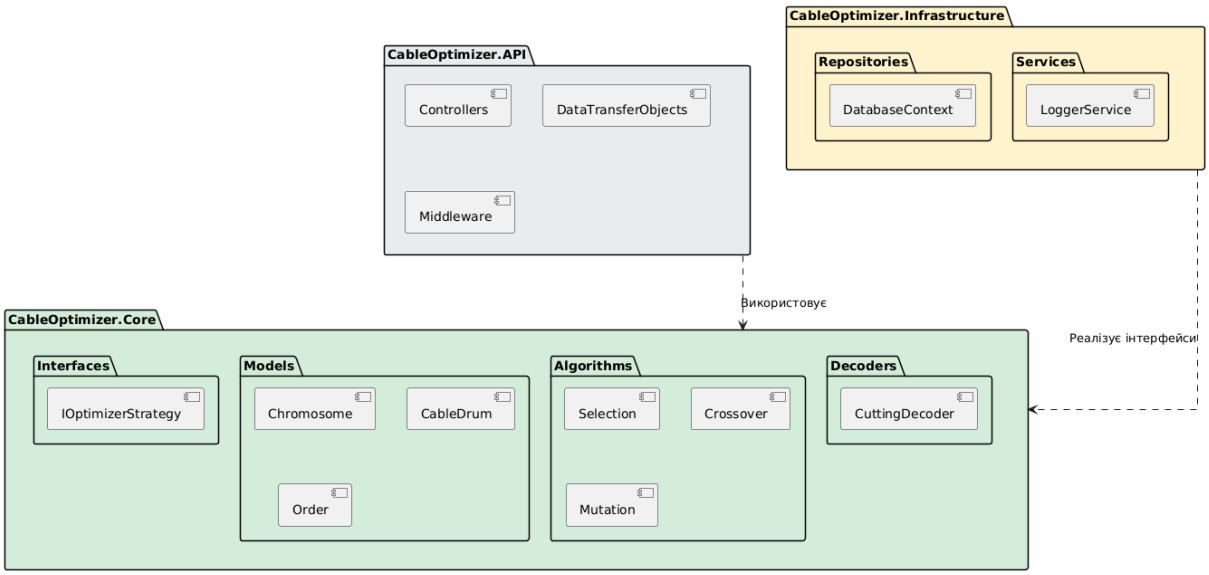


Рисунок 1 – UML-діаграма пакетів (просторів імен) програмного модуля

Центральним пакетом є ядро CableOptimizer.Core, яке не має жодних зовнішніх залежностей від інших пакетів. Воно інкапсулює виключно математичні абстракції, моделі даних (хромосоми, замовлення) та алгоритми (селекцію, кросовер, мутацію). Пакет CableOptimizer.API слугує зовнішньою оболонкою, забезпечуючи перетворення мережевих запитів у команди для ядра.

Для успішного практичного впровадження програмний модуль проектується як незалежний автономний сервіс, який взаємодіє із зовнішнім середовищем за допомогою прикладного програмного інтерфейсу на базі архітектурного стилю передачі представлення стану. Це дозволяє інтегрувати розроблене рішення у будь-яку існуючу екосистему автоматизації підприємства, незалежно від технологій, на яких вона побудована. Завдяки дотриманню архітектурного принципу слабкої пов'язаності та сильної зв'язності (модульності), математичний апарат генетичного алгоритму є повністю ізольованим від механізмів збереження даних, графічного інтерфейсу чи особливостей мережевого протоколу інтеграції. Це дозволяє проводити автономне тестування обчислювального ядра та модернізувати еволюційні оператори без ризику порушити роботу інтеграційних інтерфейсів [3].

На рис. 2 наведено детальну UML-діаграму компонентів, яка відображає внутрішню структуру рівнів програмного модуля та схему його інтеграційної взаємодії із зовнішніми складськими базами даних та робочими місцями персоналу.

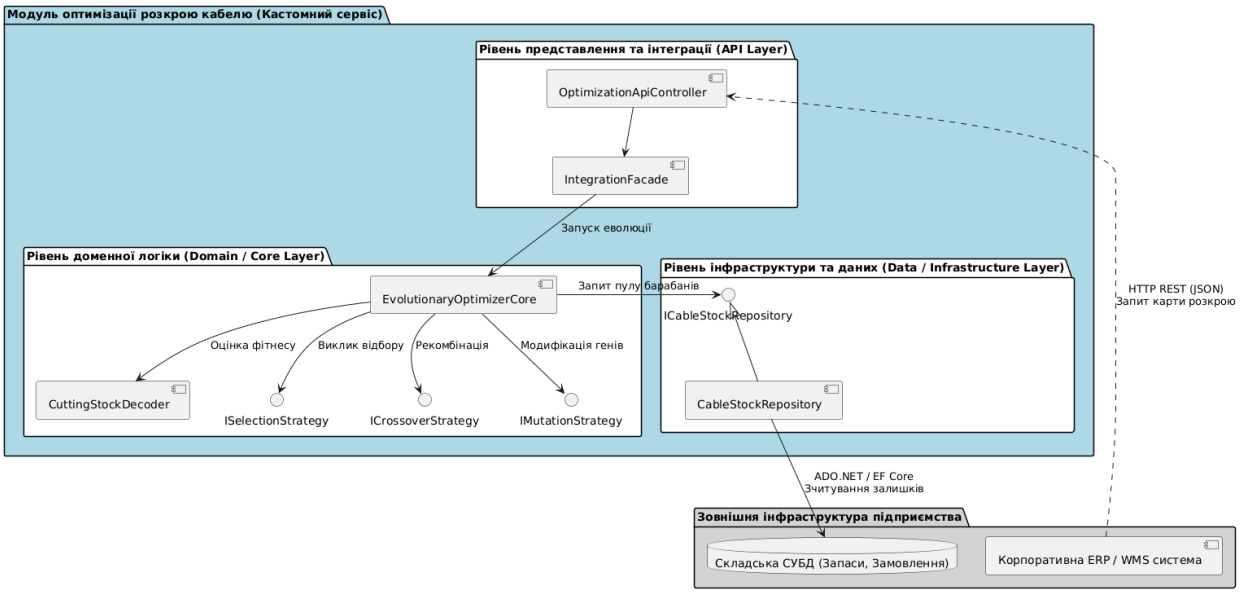


Рисунок 2 – UML-діаграма компонентів та схема інтеграції програмного модуля

Завдяки представленій компонентній структурі процес взаємодії є максимально прозорим: зовнішня система надсилає пакет поточних замовлень клієнтів у форматі JSON, рівень представлення десеріалізує ці дані, ініціює обчислювальний процес у доменному ядрі, яке, у свою чергу, через репозиторії динамічно підтягує актуальну інформацію про залишки кабельних барабанів зі складської бази даних.

Для повного опису динаміки функціонування системи, визначення часової послідовності передачі повідомлень та деталізації процесу взаємодії між різними архітектурними рівнями під час виконання процедури оптимізації розроблено UML-діаграму послідовності.

На рис. 3 наведено повний життєвий цикл виконання одного запиту на оптимізацію складських залишків, починаючи від ініціації логістом у зовнішньому інтерфейсі та закінчуючи поверненням готового результату. Ця діаграма демонструє, як послідовно виконується завантаження інформації про складські запаси, проведення ітераційних еволюційних обчислень всередині ядра та формування фінального результату. Важливою інженерною перевагою є те, що зовнішня система (WMS/ERP) повністю звільнена від виконання важких математичних розрахунків, делегуючи цю функцію розробленому модулю.

Спроектована архітектурна модель повністю відповідає стандартам розробки сучасного розподіленого програмного забезпечення та виступає надійною основою для етапу безпосереднього кодування компонентів системи.

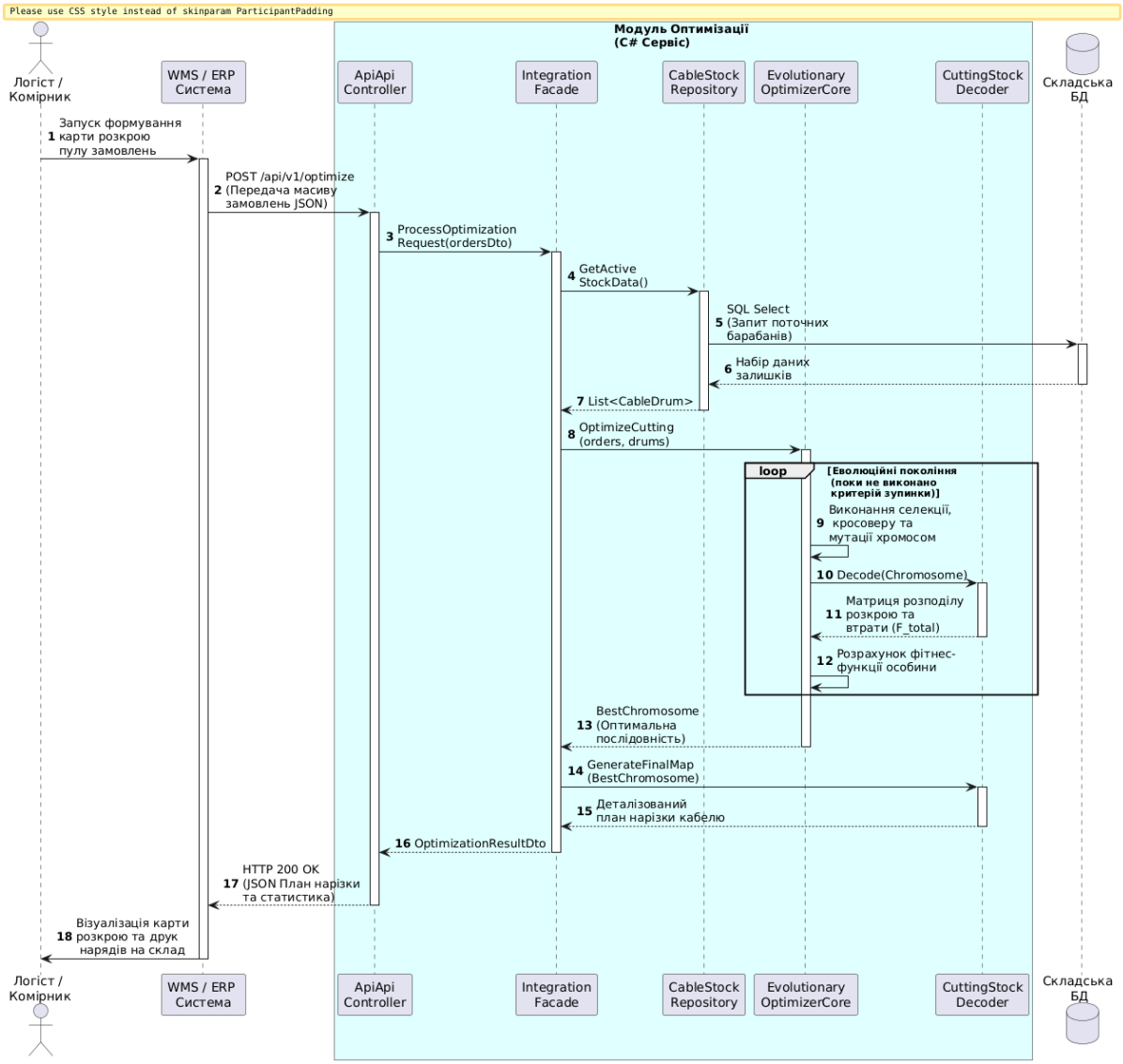


Рисунок 3 – UML-діаграма послідовності процесу інтеграційної взаємодії

Висновки. Впровадження та реалізація архітектури модуля як мікросервісу спрощує процес інтеграції складних еволюційних обчислень в існуючі WMS-системи підприємства без створення затримок у їхній роботі. Застосування оптимізаційного модуля на базі генетичних алгоритмів дозволяє досягти зниження обсягів неліквідних відходів кабелю більш ніж у 4 рази (з 9.6% до 2.3%) та суттєво зменшити потребу в розкритті нових будівельних довжин. Завдяки паралелізації обчислень, час формування оптимального плану розкрою для стрес-тесту на 500 замовлень не перевищує 15 секунд, що забезпечує високу ефективність

Список літератури

1. Araujo, S. A. D., Constantino, A. A., & Poldi, J. M. (2014). A genetic algorithm for the one-dimensional cutting stock problem with setups. Pesquisa Operacional, 34, 165-187.

2. Мартин Р. Чиста архітектура: мистецтво розробки програмного забезпечення. – Київ: Фабула, 2019. – 368 с.

3. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. – Boston: Addison-Wesley, 2023. – 560 p.

Стасів Д.Р.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЗАСОБІВ ОФЛАЙН-РОЗПІЗНАВАННЯ ТА СИНТЕЗУ МОВЛЕННЯ ДЛЯ УКРАЇНСЬКОЇ МОВИ

Вступ.Розширення ринку голосових технологій - за оцінками аналітиків, кількість голосових пристроїв у світі перевищила 8,4 млрд одиниць у 2024 році, а темп зростання сегменту становить близько 34% на рік (Grand View Research, 2024) - формує потребу у надійних, конфіденційних і незалежних від мережі рішеннях для автоматичного опрацювання мовлення. Провідні хмарні сервіси (Google Speech-to-Text, Amazon Polly, Microsoft Azure Cognitive Services) забезпечують високу якість, однак передають акустичні дані на зовнішні сервери, мають затримку 200–800 мс і не підтримують офлайн-режим для української мови.

Особливої актуальності питання офлайн-опрацювання набуває у сферах із підвищеними вимогами до конфіденційності - охороні здоров'я, юриспруденції, оборонній промисловості. За даними Сокола та Білоуса (2023), лише 12% компаній в Україні застосовують повністю автономні голосові рішення, тоді як 67% зацікавлені у переході на локальне опрацювання за умови наявності якісних інструментів. Водночас комплексний порівняльний аналіз офлайн-інструментів автоматичного розпізнавання (ASR) та синтезу мовлення (TTS) з акцентом на підтримку саме української мови залишається недостатньо представленим у вітчизняній науковій літературі, що й визначає актуальність і новизну цього дослідження.

Постановка задачі. Завданням даного дослідження є проведення комплексного порівняльного аналізу сучасних програмних засобів та бібліотек автоматичного розпізнавання (ASR) і синтезу мовлення (TTS), що функціонують у повністю автономному (офлайн) режимі. У межах роботи досліджуються їхні ключові функціональні та технічні характеристики, зокрема: точність розпізнавання, природність синтезу, затримка відповіді, вимоги до апаратного забезпечення, умови ліцензування, а також ступінь підтримки української мови. Кінцевою метою є формування обґрунтованих практичних рекомендацій щодо вибору та ефективної інтеграції цих технологій під час розробки локальних україномовних голосових інтерфейсів.

Основний матеріал. Проведений аналіз охоплює дві ключові категорії офлайн-засобів: автоматичне розпізнавання мовлення (ASR) та синтез мовлення (TTS), -із систематичною порівняльною оцінкою кожного інструменту за стандартизованими метриками.

У категорії ASR досліджено чотири основних рішення: Whisper (OpenAI), Vosk, wav2vec 2.0 (Meta) та Coqui STT. Порівняння здійснювалось за показником Word Error Rate (WER) - відсотком неправильно розпізнаних слів на україномовному тестовому корпусі. У таблиці 1 наведено стислий зміст характеристик досліджуваних систем ASR, де зіставлено їхню архітектуру, точність (WER) та основні вимоги до апаратних ресурсів.

Таблиця 1. Порівняння офлайн-засобів автоматичного розпізнавання

Інструмент	Архітектура / Підхід	Точність (WER)	Розмір / Вимоги	Швидкодія (на CPU)
Whisper (large-v3)	Трансформерна	~8%	Потребує GPU	Інференс 5–10 с на фразу
wav2vec 2.0 (Meta)	Self-supervised	~11%	~360 MB (Base)	Інференс ~1,5–2 с на фразу
Vosk	Kaldi (DNN-HMM)	~14%	~50 MB	Затримка < 1 с
Coqui STT	DeepSpeech	~22%	~180 MB	Затримка ~300–500 мс

Whisper (large-v3) демонструє найнижчий WER ~8% завдяки трансформерній архітектурі та навчанню на 680 000 годинах мультилінгвального аудіо. Модель коректно опрацьовує акцентоване мовлення, суржик і шумові умови, однак потребує значних обчислювальних ресурсів: інференс на CPU займає 5–10 секунд на фразу, що обмежує застосування без GPU-прискорення. Vosk на базі архітектури Kaldi (DNN-HMM) забезпечує WER ~14% при розмірі моделі лише 50 MB і часі відповіді менше 1 секунди на CPU. Наявна україномовна модель vosk-model-uk та ліцензія Apache 2.0 роблять Vosk оптимальним рішенням для вбудованих систем. wav2vec 2.0 (Meta) досягає WER ~11% методом самостійного навчання (self-supervised learning), що дозволяє отримувати прийнятну точність за обмеженої кількості розмічених даних -критично важлива властивість для низькоресурсних мов, зокрема для української. Coqui STT (архітектура DeepSpeech) показує WER ~22%, але надає зручний API для донавчання на спеціалізованих доменних даних.

У категорії TTS оцінювались Silero TTS, Coqui TTS, VITS/VitsUkr та eSpeak-NG. Якість синтезу вимірювалась за шкалою Mean Opinion Score (MOS) від 1 до 5. Таблиця 2 підсумовує результати оцінювання інструментів TTS, демонструючи залежність між природністю згенерованого мовлення (MOS), архітектурою та ресурсоємністю кожної моделі.

Таблиця 2. Порівняння офлайн-засобів синтезу мовлення (TTS)

Інструмент	Архітектура / Метод	Оцінка MOS	Розмір / Пам'ять	Швидкодія (на CPU)
Silero TTS	Власна E2E (CNN + Вокодер)	4,3/5	~55 MB	Інференс ~30–50 мс
VITS/VitsUkr	VITS	3,9/5	~140 MB	Інференс ~500–800 мс
Coqui TTS	Tacotron2 + HiFi-GAN	3,8/5	~180 MB	Інференс > 1 с
eSpeak-NG	Формантний синтез	2,1/5	~5 MB	Затримка < 10 мс

Silero TTS отримав найвищу оцінку MOS 4,3/5 при компактному розмірі моделі (~55 MB) і високій швидкості синтезу; ключовою перевагою є природне інтонування й наголосування в україномовній моделі. Coqui TTS (Tacotron2 + HiFi-GAN) демонструє MOS 3,8/5 з можливістю клонування голосу, але потребує ~180 MB пам'яті. Архітектура VITS (Variational Inference with adversarial learning), реалізована у проєкті VitsUkr, забезпечує MOS 3,9/5 завдяки поєднанню акустичної моделі та вокодера в єдиній мережі, що забезпечує природніший ритм мовлення. eSpeak-NG на основі формантного синтезу показує MOS 2,1/5, однак його розмір (~5 MB) та миттєва швидкість роблять його єдиним придатним варіантом для мікроконтролерів та IoT-пристроїв.

На основі комплексного аналізу сформовано рекомендовані конфігурації для систем різного класу. Для вбудованих систем (Raspberry Pi 4 і подібних) оптимальним є стек Vosk + eSpeak-NG (~200 MB, час відповіді < 50 мс). Для настільних систем і серверів рекомендується Vosk + Silero TTS (~500 MB RAM) як найкращий баланс точності, природності й ресурсоємності. У застосунках із критичними вимогами до точності (медицина, юриспруденція) доцільне використання Whisper large-v3 + Silero TTS за наявності GPU з об'ємом відеопам'яті від 8 GB. Практичне тестування базової конфігурації на Raspberry Pi 4 (4 GB RAM) підтвердило її працездатність: витрата оперативної пам'яті -310 MB, час ініціалізації -1,1 с, середній час інференсу -19 мс, точність розпізнавання голосових команд - 95,2%.

Поточними обмеженнями розвитку офлайн-рішень для української мови є недостатній обсяг якісних акустичних датасетів (корпус Mozilla CommonVoice UA нараховує ~85 годин запису проти 2 800+ годин для англійської мови), відсутність охоплення діалектної варіативності та підвищена морфологічна складність мови. Разом із тим динаміка розвитку є позитивною: за 2023–2025 роки обсяг CommonVoice UA зріс утричі, а квантизовані версії Whisper (Whisper.cpp, 4-bit quantization) прискорюють інференс на CPU до 10 разів без суттєвої втрати точності.

Висновок. Результати проведеного аналізу свідчать, що побудова повноцінного офлайн-голосового інтерфейсу з підтримкою української мови є технічно реалізованою задачею на сучасному рівні відкритого програмного забезпечення. Встановлено, що Vosk забезпечує оптимальне співвідношення точності (WER ~14%) і продуктивності на CPU; Silero TTS демонструє найвищу природність синтезованого мовлення (MOS 4,3/5) серед безкоштовних україномовних рішень; для задач із критичними вимогами до точності доцільне застосування Whisper large-v3 за умови GPU-прискорення. Запропоновані конфігурації можуть бути використані при розробці голосових асистентів, систем «розумного дому», медичних інформаційних систем та оборонних застосунків, де передача акустичних даних на зовнішні сервери є неприпустимою.

Список літератури

1. Сокол І., Білоус М. (2023). Застосування технологій розпізнавання мовлення в україномовних інформаційних системах: сучасний стан і перспективи. Вісник Національного університету «Львівська політехніка», серія: Інформаційні системи та мережі, 4(2), 88–102.
2. Baevski A., Zhou H., Mohamed A., Auli M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. Advances in Neural Information Processing Systems (NeurIPS), 33, 12449–12460.
3. Radford A., Kim J.W., Xu T. et al. (2023). Robust Speech Recognition via Large-Scale Weak Supervision. Proceedings of the 40th International Conference on Machine Learning (ICML), 202, 28492–28518.
4. Kim J., Kong J., Son J. (2021). Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. Proceedings of ICML, 139, 5530–5540.
5. Grand View Research (2024). Voice Assistant Market Size, Share & Trends Analysis Report 2024–2030. Grand View Research Inc.

Ситник Назар Дмитрович
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ, ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ ВИЗНАЧЕННЯ ХАРАКТЕРИСТИК ПРОДУКТІВ З ДОПОМОГУЮ АНАЛІЗУ ЗОБРАЖЕНЬ

Вступ. Проблема контролю харчування набуває дедалі більшої актуальності: за даними МОЗ України, близько 60 % дорослого населення мають ознаки ожиріння, що безпосередньо пов’язано з розвитком діабету та серцево-судинних захворювань [1]. Аналіз публікацій останніх років (Tan & Le, 2019; Goodfellow et al., 2020; Chollet, 2021) свідчить про стрімкий розвиток методів глибокого навчання та комп’ютерного зору, що відкривають можливості для автоматизації аналізу харчування. Існуючі мобільні застосунки — MyFitnessPal, FatSecret, Yazio — переважно покладаються на ручне введення даних або сканування штрих-кодів, що обмежує їх застосовність для домашньої та ресторанної їжі і займає в середньому 4,8 хвилини на один прийом їжі [2]. Найперспективнішим підходом є використання згорткових нейронних мереж (CNN) для автоматичного розпізнавання продуктів харчування за фотографією, проте якість їх роботи критично залежить від попередньої обробки вхідних зображень, що і зумовлює актуальність даного дослідження.

Постановка задачі. Об’єкт дослідження – процес попередньої обробки зображень у системах автоматичного розпізнавання продуктів харчування на основі глибокого навчання. Предмет дослідження – алгоритмічний склад, послідовність та параметри операцій конвеєра попередньої обробки зображень для забезпечення стабільної роботи моделі EfficientNet-B0. Мета – розробити та обґрунтувати ефективний алгоритм попередньої обробки зображень, що забезпечує максимальну точність класифікації продуктів харчування в умовах різноманітності характеристик мобільних знімків.

Основний матеріал. Розроблений конвеєр попередньої обробки зображень складається з шести послідовних операцій, реалізованих засобами бібліотек OpenCV та NumPy на серверній

стороні. Розміщення логіки обробки на сервері забезпечує однаковість результатів незалежно від типу мобільного пристрою та спрощує подальше оновлення без необхідності оновлення клієнта.

Перший крок – декодування та валідація: вхідний байтовий потік перетворюється на масив NumPy за допомогою функції cv2.imdecode() з одночасною перевіркою коректності (тривимірність масиву, мінімальний розмір 100×100 пікселів). Другий крок – конвертація колірного простору BGR → RGB: бібліотека OpenCV за замовчуванням зчитує зображення у форматі BGR, тоді як модель EfficientNet-B0 навчена на RGB-зображеннях датасету ImageNet; невідповідність порядку каналів спотворює активації перших шарів CNN навіть при якісно навченій моделі. Третій крок – масштабування до розміру 224×224 пікселів за допомогою білінійної інтерполяції (INTER-LINEAR), що є стандартним вхідним форматом EfficientNet-B0. Четвертий крок – шумозаглушення фільтром Гауса з ядром 3×3: знімки мобільних камер містять цифровий шум матриці сенсора, а гаусівська фільтрація прибирає дрібні артефакти, зберігаючи форму, колір і текстуру їжі – основні ознаки для класифікатора. П'ятий крок – нормалізація: ділення на 255,0 переводить цілочисельні значення пікселів з діапазону [0; 255] у дійсночисельний [0; 1], що стабілізує градієнтний спуск і відповідає умовам передавчання на ImageNet. Шостий крок – формування пакетного тензора: функція np.expand_dims() перетворює масив розмірності [224, 224, 3] на тензор [1, 224, 224, 3], готовий до подачі на вхід нейронної мережі.

Для класифікації зображень застосовано архітектуру EfficientNet-B0 [3], що базується на принципі комплексного масштабування (compound scaling). Її ключова перевага – лише 5,3 млн параметрів проти 25 млн у ResNet-50 при порівнянній точності, що дозволяє розгорнути модель у хмарному середовищі. Навчання виконувалось методом двофазного Transfer Learning на датасеті Food-101 (101 клас, 101 000 зображень): у фазі 1 навчалась лише нова класифікаційна голова при замороженій базовій моделі (Adam, lr = 1e-3, 10–15 епох); у фазі 2 – донавчались верхні 20 шарів зі зменшеним кроком навчання (lr = 1e-5) для запобігання катастрофічному забуванню низькорівневих ознак ImageNet.

Для підвищення надійності в реальних умовах використання впроваджено механізм порогового відсіювання: якщо максимальне значення вихідного вектора Softmax не перевищує 0,5, система повідомляє користувача про низьку впевненість у результаті та пропонує повторити фотографування. Такий підхід виключає хибну видачу КБЖВ при нечіткому або нетиповому знімку.

Висновки. Розроблено шестикроковий алгоритм попередньої обробки зображень (decode → BGR → RGB → resize 224×224 → GaussianBlur → normalize → batch tensor), що є ключовим компонентом системи автоматичного визначення калорійності продуктів харчування за фотографією. Наукова новизна полягає в обґрунтуванні оптимального складу та послідовності операцій конвеєра з урахуванням специфіки мобільних камер та вимог архітектури EfficientNet-B0. Практичне значення: час введення даних про один прийом їжі скорочується у 4 рази – з 4,8 до 1,2 хвилини порівняно з ручним введенням; розміщення конвеєра на серверній стороні забезпечує незалежність від типу клієнтського пристрою та спрощує подальший розвиток системи.

Список літератури

1. МОЗ України. (2023). Статистичний збірник «Здоров'я населення України». Київ: МОЗ України.

2. Бойко І., Семенченко В. (2023). Огляд мобільних застосунків для контролю харчування: функціональний аналіз та перспективи. Комп'ютерні науки та інформаційні технології, 18(2), 45–57.

3. Tan M., Le Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th ICML, 6105–6114.

4. Chollet F. (2021). Deep Learning with Python. 2nd Edition. Manning Publications.

5. Goodfellow I., Bengio Y., Courville A. (2020). Deep Learning. 2nd ed. MIT Press.

Гриник Н.М.
аспірант 3 курсу Навчально-наукового інституту фізико-технічних та комп'ютерних наук
Чернівецького національного університету імені Юрія Федьковича
Науковий керівник д.т.н., доцент Баловсяк С.В., кафедра комп'ютерних систем та мереж,
Чернівецький національний університет імені Юрія Федьковича

ПОБУДОВА ТРИВИМІРНИХ МОДЕЛЕЙ ОБ'ЄКТІВ МЕТОДОМ ФОТОГРАММЕТРІЇ З КОЛЬОРОВОЮ КОРЕКЦІЄЮ ЗОБРАЖЕНЬ

Вступ. У сучасних комп'ютерних системах тривимірні (3D) моделі об'єктів застосовуються при вирішенні багатьох прикладних задач. До сфер застосування 3D моделей належить архітектура, будівництво, відеоігри, медицина, освіта, транспорт та інші галузі. Побудова 3D моделей часто виконується методом фотограмметрії на основі серії цифрових зображень об'єктів, отриманих за допомогою відеокамер [1]. Метод фотограмметрії реалізовано в спеціалізованих програмах, наприклад, в 3DF Zephyr [2]. Проте, в процесі отримання серії зображень на різних кадрах кольоровий спектр об'єкта може відрізнятися, оскільки в сучасних відеокамерах виконується автоматичний баланс білого (Auto White Balance, AWB). Також спектр об'єкта може змінюватися за рахунок зміни освітлення та попадання в кадр сторонніх предметів, що негативно впливає на точність побудованих 3D моделей. Тому завдання кольорової корекції зображень перед побудовою 3D моделей методом фотограмметрії є актуальним, оскільки дає змогу краще суміщати кадри і підвищити точність моделей.

Постановка задачі. Метою даної роботи є побудова 3D моделей об'єктів методом фотограмметрії в програмі 3DF Zephyr із кольоровою корекцією зображень. Для корекції зображень потрібно розробити програму на мові Python, яка б виконувала сегментацію досліджуваних об'єктів на зображеннях нейронною мережею з архітектурою YOLO (You Only Look Once). Корекцію кольору реалізувати шляхом встановлення однакових значень для каналів червоного, зеленого та синього кольорів для сегменту досліджуваного об'єкта на різних кадрах.

Основний матеріал. За допомогою розробленої програми на мові Python зчитано початкові зображення, виділено на них сегменти досліджуваних об'єктів згортковою нейронною мережею YOLO (версії 8, середнього розміру) [3, 4]. Для сегменту об'єкту на всіх зображеннях встановлено однакові відносні значення для червоного, зеленого та синього каналу кольору. На основі коректованих зображень побудовано 3D модель в програмі 3DF Zephyr (рис. 1).

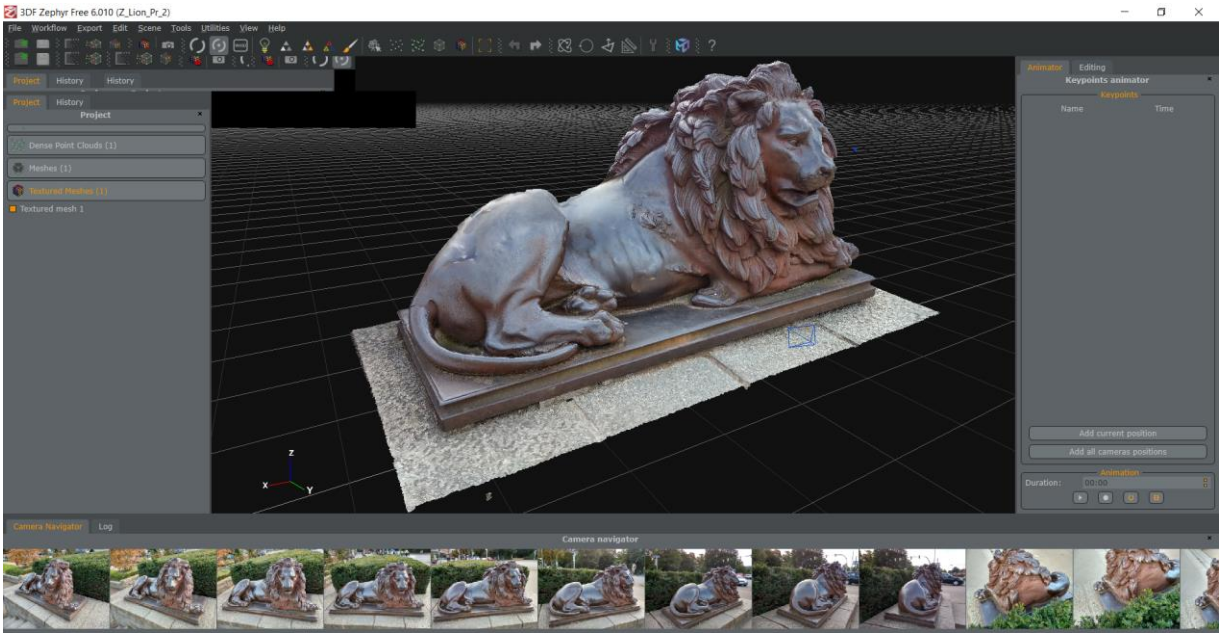


Рисунок 1 – Тривимірна модель об'єкта, побудована в програмі 3DF Zephyr

Процес побудови 3D моделі складався з 4 етапів, у результаті виконання яких створено: 1) розріджену хмару точок (Sparse Point Cloud); 2) щільну хмару точок (Dense Point Cloud); 3) сітку полігонів (Mesh); 4) текстурну сітку (Textured Mesh). За рахунок корекції кольору для вхідних зображень отримано тривимірну модель об'єкта (статуї лева) без значних спотворень (рис. 1). У процесі побудови моделі в програмі 3DF Zephyr використано два режими: 1) режим за замовчуванням з середньою деталізацією об'єкта (Default) (рис. 2а); 2) режим з високою деталізацією (High) (рис. 2б). У режимі з високою деталізацією час побудови збільшився більш ніж у 2 рази, проте значно збільшилася кількість вершин і трикутників тріангуляційної сітки, а відповідно підвищено точність та візуальну якість 3D моделі.

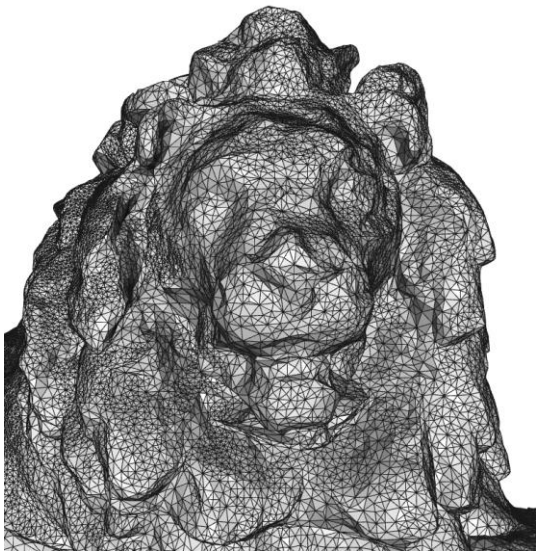
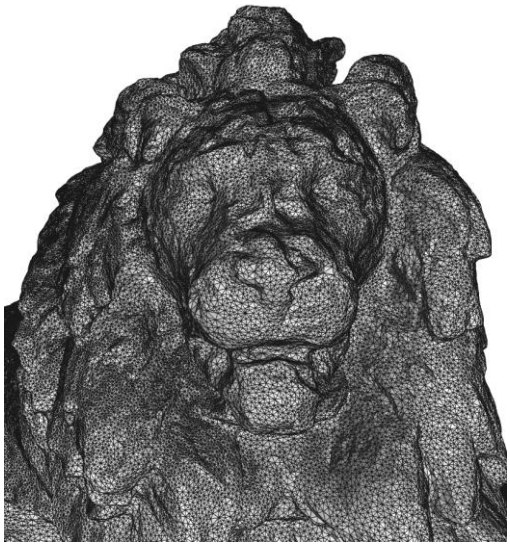
а)  б) 

Рисунок 2 – Фрагменти тріангуляційних сіток для побудованих тривимірних моделей у режимах побудови: а) за замовчуванням (117 911 вершин сітки, 169 556 трикутників); б) з високою деталізацією (458 449 вершин, 636 244 трикутників)

Побудовані в програмі 3DF Zephyr тривимірні моделі експортуються у файли різних форматів (наприклад, у формат Obj). Такі файли 3D моделей можуть зчитуватися, зокрема, програмами на мові Python та програмою Blender з метою їх аналізу та обробки (децимації, згладжування та ін.).

Висновки. Розроблено програму на мові Python для сегментації зображень нейронною мережею YOLO та кольорової корекції виділених сегментів. За рахунок корекції початкових зображень підвищено точність побудови тривимірних моделей об'єктів методом фотограмметрії в програмі 3DF Zephyr. Тривимірні моделі побудовано в режимах з середньою та високою деталізацією, що дозволяє вибирати 3D модель для конкретної задачі з урахуванням кількості вершин і полігонів для її тріангуляційної сітки. Отримані 3D моделі об'єктів можуть застосовуватися в системах тривимірної графіки, доповненої та віртуальної реальностей.

Список літератури

1. Sh. Zhang, Ch. Liu, N. Haala, "Guided by model quality: UAV path planning for complete and precise 3D reconstruction of complex buildings," *International Journal of Applied Earth Observation and Geoinformation*, Vol. 127 (103667), 2024, pp. 1-13.

2. 3DF Zephyr. The Complete Photogrammetry Solution. URL: https://www.3dflow.net.

3. S.-H. Tsang, "Review: YOLOv8 (Object Detection). Outperforms YOLOv5, YOLOv6 and YOLOv7," URL: https://sh-tsang.medium.com/review-yolov8-object-detection-5214fa105731.

4. N. Hrynyk, S. Balovsyak, V. Branashko, O. Dubovyk, Kh. Odaiska, "Improving accuracy of photogrammetry method by masking images and using coordinates of video cameras," *Proceedings of SPIE: The International Society for Optical Engineering*, 2025, Vol. 13813, pp. 138130L-1 – 138130L-7.

Яковлєв С.В.
аспірант 3 курсу Навчально-наукового інституту фізико-технічних та комп'ютерних наук
Чернівецького національного університету імені Юрія Федьковича
Науковий керівник д.т.н., доцент Баловсяк С.В., кафедра комп'ютерних систем та мереж,
Чернівецький національний університет імені Юрія Федьковича

ЗГЛАДЖУВАННЯ ТРИВИМІРНИХ МОДЕЛЕЙ ОБ'ЄКТІВ ФІЛЬТРОМ ЛАПЛАСА

Вступ. На даний час тривимірні (3D) моделі об'єктів отримали широке застосування в науці, техніці, освіті, медицині, промисловості, будівництві, системах доповненої реальності (Augmented Reality) та віртуальної реальності (Virtual reality) [1]. Проте, на тривимірних моделях об'єктів, отриманих, зокрема, методом фотограмметрії, можуть бути присутніми шуми та дефекти. Такі спотворення 3D моделей ускладнюють їх подальшу обробку. Тому завдання згладжування 3D моделей шляхом фільтрації є актуальним, оскільки у результаті такої обробки значно зменшується рівень шуму на зображенні.

Постановка задачі. Згладжування тривимірних моделей об'єктів полягає у зміні *xyz* координат вершин для їх тріангуляційної сітки (mesh) таким чином, щоб зменшити випадкові відхилення вершин відносно сусідніх (які з'єднуються ребрами сітки). Метою даної роботи є розробка програмного забезпечення на мові Python для згладжування тріангуляційної сітки фільтром Лапласа. Для обробки 3D моделей потрібно вибрати такі параметри фільтру Лапласа, при яких шум тріангуляційної сітки у значній мірі зменшується, але при цьому відсутнє надмірне згладжування форми досліджуваних об'єктів.

Основний матеріал. Важливим фільтром, який застосовується при згладжуванні 3D моделей, є фільтр Лапласа (Laplacian filter). Дія фільтра Лапласа полягає в ітераційному згладжуванні тріангуляційної сітки шляхом усереднення *xyz* координат кожної вершини з координатами сусідніх вершин.

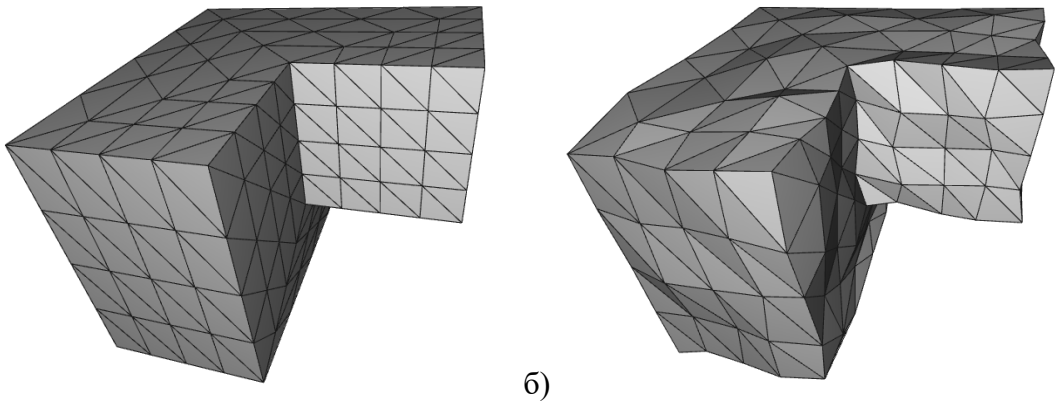
Для згладжування 3D моделей розроблено програмне забезпечення на мові Python з використанням бібліотеки «open3d» [1], у якій фільтр Лапласа реалізується функцією

«filter_smooth_laplacian(Q_{it}, λ_L)»,

де Q_{it} – кількість ітерацій, λ_L– коефіцієнт сили фільтра ($0 < \lambda_L < 1$).

У порівнянні з найпростішим усереднюючим фільтром (Average filter) фільтр Лапласа менше згладжує гострі кути і таким чином зберігає форму об'єкта.

Дослідження можливостей фільтру Лапласа виконано на прикладі модельованої 3D моделі простої форми, яка є поєднанням двох прямокутних паралелепіпедів (рис. 1а). 3D модель створено засобами системи OpenScad, а додаткові вершини тріангуляційної сітки додано функцією «subdivide_midpoint()» бібліотеки «open3d». Спотворення, які можуть виникати на 3D моделях реальних об'єктів при їх побудові методом фотограмметрії або з використанням лазерних далекомірів, моделювалися рівномірно розподіленим шумом (рис. 1б).



а) б)

Рисунок 1 – Тривимірні моделі: а) початкова; б) з рівномірно розподіленим шумом у діапазоні 0.15; кількість вершин тріангуляційної сітки QV =194, кількість трикутників сітки QT=384

Після згладжування 3D моделі фільтром Лапласа отримано такі результати (рис. 2). При збільшенні коефіцієнта сили фільтра λ_L згладжування тріангуляційної сітки зростає. Більше згладжування також можна отримати шляхом збільшення кількості ітерацій Q_{it} (рис. 2г).

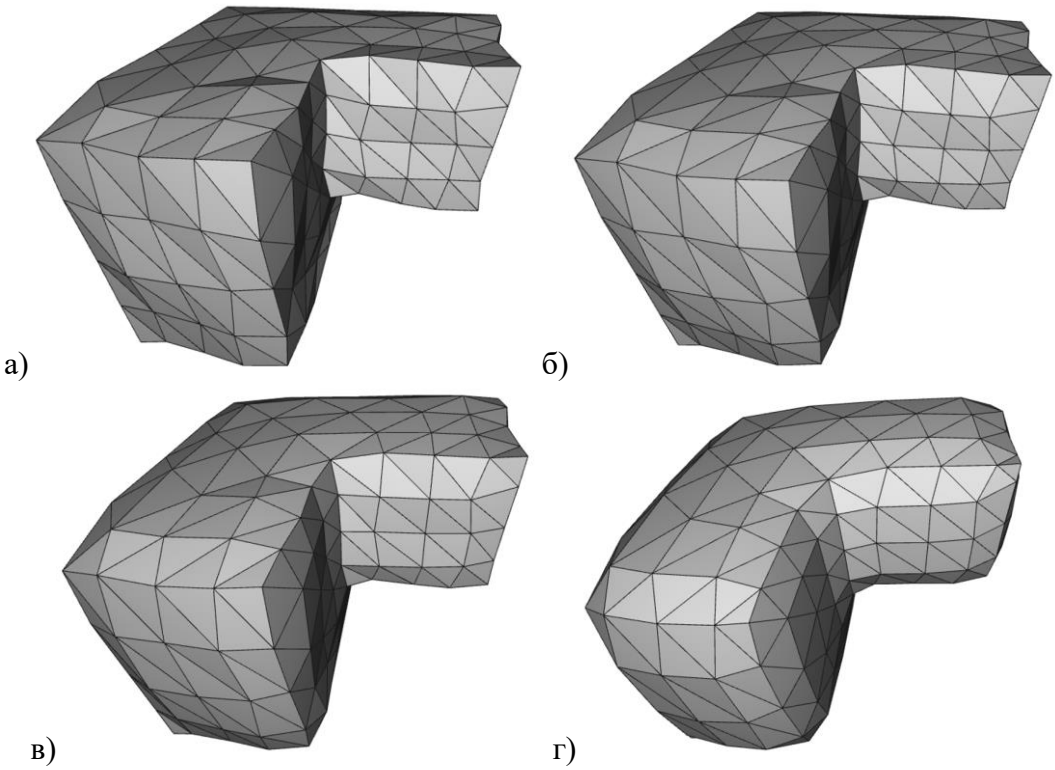


Рисунок 2 – Тривимірні моделі після фільтрації Лапласа: а) λ_L= 0.1, Q_{it} = 2; б) λ_L= 0.2, Q_{it} = 2; в) λ_L= 0.3, Q_{it} = 2; г) λ_L= 0.3; Qit = 5

Серед результатів фільтрації найвищу візуальну якість досліджуваної 3D моделі отримано для коефіцієнта сили фільтра λ_L= 0.2 (рис. 2б), при якому шум сітки значно зменшено, але відсутнє надмірне згладжування сітки (рис. 2в, рис. 2г). Оцінку результатів фільтрації симульованих 3D моделей також можливо виконувати за об'єктивними критеріями, зокрема, за коренем середньої квадратичної помилки (Root Mean Square Error – RMSE) для координат вершин фільтрованої моделі відносно координат вершин початкової моделі (без шуму).

Висновки. Згладжування тривимірних моделей об'єктів фільтром Лапласа дозволяє зменшити рівень шуму таких моделей та підвищити їх візуальну якість. Шляхом вибору коефіцієнта сили фільтра λ_L досягається компроміс між зменшенням рівня шуму та згладжуванням форми об'єкта. За допомогою розробленої програми на мові Python можливо згладжувати складні моделі реальних об'єктів. Для підвищення візуальної якості згладжених 3D моделей також доцільно застосовувати складніші фільтри, зокрема, фільтр Таубіна (Taubin filter) [2] та білатеральний фільтр [3] .

Список літератури

1. Open3D. URL: https://www.open3d.org.

2. A.M. Esmoris, X. Garcia-Martinez, M. Ladra, J.C. Cabaleiro, and F.F. Rivera, "Introducing the Taubin-Weingarten algorithm to compute second-order geometric descriptors in 3D point clouds," *Journal of Computational and Applied Mathematics*, Vol. 485 (117569), 2026, pp. 1-19.

3. S. Yakovliev, S. Balovsyak, M. Borcha, I. Iakovlieva, and I. Yanchuk, "Automatic bilateral filtering of digital images using histograms," *Proceedings of SPIE: Seventeenth International Conference on Correlation Optics*, 2025, Vol. 13813, pp. 138130M-1 – 138130M-6.

Гевко Д. З.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ АВТОМАТИЧНОГО ГЕНЕРУВАННЯ СУБТИТРІВ ТА ОНЛАЙН ПЕРЕКЛАДУ ВІДЕО

Вступ. У сучасному цифровому середовищі відеоконтент стає основним способом передачі знань і комунікації. Платформи дистанційного навчання, новинні ресурси та корпоративні системи щоденно генерують мільярди хвилин відеозаписів різними мовами. Разом із тим, особи з вадами слуху, а також аудиторія, яка не володіє мовою оригіналу, стикаються з проблемою недоступності такого контенту. Ручне субтитрування залишається трудомістким процесом: транскрибування однієї години відео займає фахівця від 4 до 10 годин роботи, тоді як вартість послуг професійного перекладача субтитрів суттєво обмежує можливості малих організацій [1, 2]. Актуальність дослідження зумовлена потребою у доступному інструменті автоматичного субтитрування, що поєднує розпізнавання мовлення та машинний переклад в єдиному десктопному застосунку.

Постановка задачі. Об'єкт дослідження – процес автоматизованого генерування субтитрів для відеоконтенту. Предмет дослідження – засоби проектування програмного модуля на основі нейронної моделі розпізнавання мовлення Whisper та API машинного перекладу. Мета дослідження – розробити програмний модуль SubGen AI для автоматичного генерування субтитрів у форматах SRT та VTT та їх онлайн перекладу на цільову мову, реалізований у вигляді десктопного застосунку з графічним інтерфейсом.

Основний матеріал. Застосунок SubGen AI реалізовано мовою Python 3.11 з використанням двошарової архітектури: шар ядра (core) та шар графічного інтерфейсу (gui). Графічний інтерфейс побудовано на стандартній бібліотеці tkinter, що забезпечує автономну роботу без підключення до зовнішнього сервера. Загальна архітектура модуля наведена на рисунку 1.

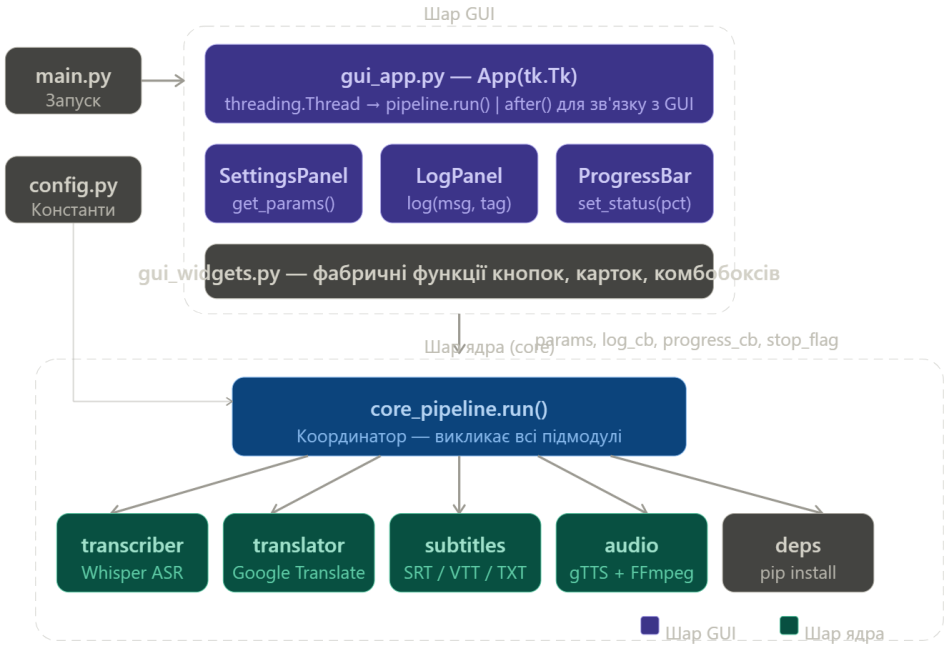


Рисунок 1 – Схема взаємодії модулів застосунку SubGen AI

Шар ядра складається із шести модулів: core_pipeline (координатор пайплайну), core_transcriber (розпізнавання мовлення через Whisper), core_translator (переклад через GoogleTranslator), core_subtitles (генерація SRT/VTT/TXT), core_audio (TTS-синтез та мікшування через gTTS і FFmpeg) та core_deps (автоматична перевірка і встановлення залежностей). Центральна функція run() модуля core_pipeline приймає словник параметрів та

три callback-функції: log_cb, progress_cb та stop_flag, що забезпечує коректне переривання між кроками обробки.

Ключовим компонентом системи є модуль core_transcriber.py, що реалізує розпізнавання мовлення на базі моделі Whisper від OpenAI [3]. Whisper навчено на 680 000 годинах різномовного мовлення та підтримує 99 мов без додаткового дообнавчання. Модель приймає відеофайл безпосередньо завдяки вбудованій підтримці FFmpeg, без попередньої екстракції аудіодоріжки. Результатом є масив сегментів з полями *start*, *end* та *text*. Застосунок підтримує п'ять конфігурацій моделі (tiny, base, small, medium, large), що дозволяє балансувати між точністю розпізнавання та швидкістю обробки залежно від апаратних можливостей. Порівняння конфігурацій моделей наведено у таблиці 1.

Таблиця 1 – Порівняння конфігурацій моделей Whisper

Модель	Параметри	Відносна швидкість	WER (EN)	Потреби VRAM
tiny	39 M	32×	~5.7%	~1 ГБ
base	74 M	16×	~4.2%	~1 ГБ
small	244 M	6×	~3.0%	~2 ГБ
medium	769 M	2×	~2.9%	~5 ГБ
large	1 550 M	1×	~2.7%	~10 ГБ

Переклад субтитрів реалізовано у модулі core_translator.py за допомогою бібліотеки *deep-translator*, що надає уніфікований інтерфейс до Google Translate API без необхідності окремого API-ключа. Алгоритм обходить масив сегментів у циклі: для кожного сегмента текст передається до *GoogleTranslator.translate()*, а результат записується у поверхневу копію сегмента (*copy.copy()*) зі збереженням незмінними полів *start* та *end*. Завдяки цьому таймкоди оригіналу і перекладу збігаються на 100%. При виникненні помилки (мережевий таймаут, порожній рядок) сегмент зберігається з оригінальним текстом, що гарантує відсутність пропусків у файлі субтитрів.

Модуль core_subtitles.py реалізує генерування субтитрів у трьох форматах. Функція segments_to_srt() формує нумеровані блоки з таймкодами у стандарті ГГ:ХХ:СС,МС. Функція segments_to_vtt() генерує файл формату WebVTT із заголовком WEBVTT, використовуючи крапку замість коми у таймкодах відповідно до стандарту W3C [4]. Функція segments_to_txt() зберігає суцільний текст без таймкодів. Для кожного обраного формату зберігаються два файли: оригінальна версія (суфікс _orig) та перекладена (суфікс з кодом мови, наприклад _uk.srt).

Опціональний модуль core_audio.py забезпечує синтез озвученої аудіодоріжки. Функція build_dubbed_audio() створює порожній numpy-буфер тривалістю відео, синтезує MP3 для кожного сегмента через gTTS, конвертує у WAV через FFmpeg та розміщує у буфері точно за таймкодами. Якщо синтезоване аудіо довше за відведений часовий слот, застосовується ресемплінг через scipy.signal.resample. Функція mix_audio_with_ffmpeg() підтримує три режими мікшування: replace (лише озвучення), mix_quiet (озвучення + тихий оригінал 15%) та mix_equal (рівне змішування). Детальна схема пайплайну наведена на рисунку 2.

Тестування якості розпізнавання мовлення проводилось на шести відеоматеріалах різних типів на апаратному забезпеченні з GPU NVIDIA RTX 3060 (6 ГБ VRAM). Оцінка виконувалась за метрикою WER (Word Error Rate) — часткою слів з помилками відносно еталонного тексту. Порівняння результатів для моделей small та large наведено у таблиці 2.

Модель large забезпечує WER на 2–4.5% нижчий порівняно з small для всіх типів контенту. Найвищий WER спостерігається для мовлення з нестандартним акцентом (13.2% для small) — через фонетичні відхилення від навчальних даних. Переклад субтитрів зберігає таймкоди оригіналу на рівні 100% у всіх тестових сценаріях, а час перекладу 100 сегментів не перевищує 25 секунд при стабільному інтернет-з'єднанні [5].

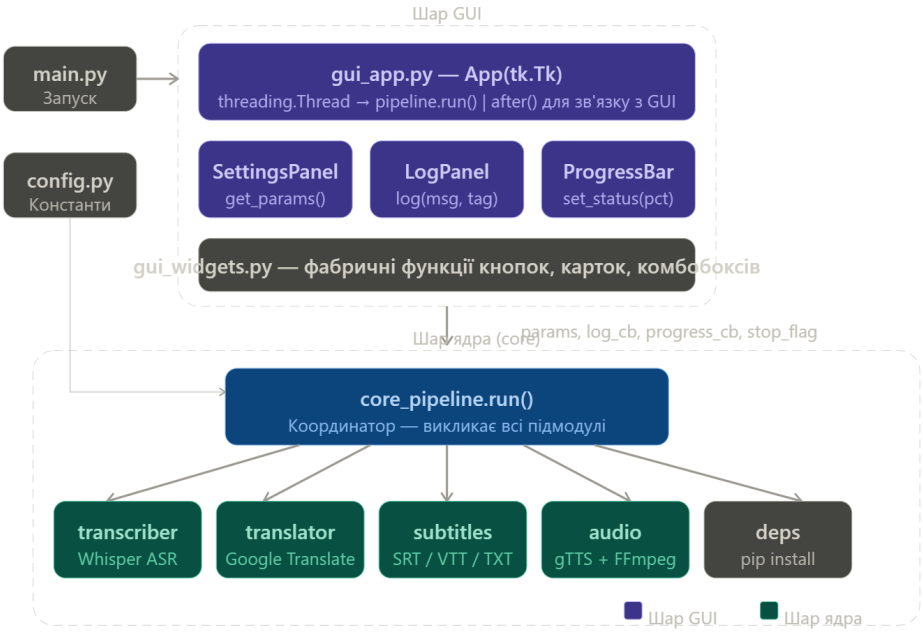


Рисунок 2 – Узагальнений алгоритм пайплайну генерування субтитрів

Таблиця 2 – Результати тестування якості розпізнавання мовлення

Тип відео	Мова	Тривалість	WER small, %	WER large, %	Час (small)
Навчальний відеоурок	EN	10 хв	5.8	3.1	1.2 хв
Розмовний подкаст	EN	15 хв	9.4	6.2	1.8 хв
Новинний сюжет	UK	5 хв	8.1	5.4	0.6 хв
Академічна лекція	EN	20 хв	6.3	3.8	2.4 хв
Інтерв'ю (акцент)	EN	8 хв	13.2	8.7	1.0 хв
Художній фільм	EN	12 хв	11.7	7.9	1.5 хв

Висновки. У рамках роботи розроблено програмний модуль SubGen AI для автоматичного генерування субтитрів та онлайн перекладу відео. Двошарова архітектура (core + gui) забезпечує незалежність бізнес-логіки від інтерфейсу та можливість розширення функціоналу. Модель Whisper підтвердила ефективність розпізнавання для 99 мов із WER від 3.1% (академічна лекція, модель large) до 13.2% (акцентоване мовлення, модель small). Алгоритм посегментного перекладу гарантує 100% збереження таймкодів оригіналу. Додаткова функція TTS-озвучення розширює застосунок до повноцінного дублювального інструменту. Практичне застосування системи є важливим для платформ дистанційного навчання, медіа-верифікації та забезпечення доступності контенту для осіб з вадами слуху.

Список літератури

1. Antonakaki P. et al. A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks // Expert Systems with Applications. 2021. Vol. 164. P. 114006.

2. Radford A. et al. Robust Speech Recognition via Large-Scale Weak Supervision // arXiv preprint arXiv:2212.04356. 2022.

3. OpenAI. Whisper: Automatic Speech Recognition. URL: https://github.com/openai/whisper.

4. W3C. Web Video Text Tracks Format (WebVTT). URL: https://www.w3.org/TR/webvtt1/.

5. deep-translator. Python library for Google Translate API. URL: https://github.com/nidhaloff/deep-translator.

6. gTTS. Google Text-to-Speech Python library. URL: https://github.com/pndurette/gTTS.

7. FFmpeg Documentation. URL: https://ffmpeg.org/documentation.html.

Мамчур Т. Б.
аспірант 3 курсу ІКНІ НУ ЛП
Науковий керівник к.т.н. Опотяк Ю.В., кафедра АСУ НУ ЛП

ПІДВИЩЕННЯ ШВИДКОДІЇ ІНТЕЛЕКТУАЛЬНИХ ЗАСОБІВ МОБІЛЬНИХ РОБОТОТЕХНІЧНИХ ПЛАТФОРМ.

Вступ. Сучасний етап розвитку мобільних робототехнічних систем (МРС) орієнтований на широке використання інтелектуальних та нейрокомп'ютерних засобів. Вони застосовуються для оцінювання даних, які надходять із давачів в умовах завад, розпізнавання зображень і сцен, а також для знаходження оптимального маршруту в невизначеному ландшафті. Критичною вимогою до таких платформ є опрацювання різних за інтенсивністю потоків даних у реальному часі. При цьому апаратні засоби повинні задовольняти жорсткі обмеження щодо габаритів та енергоспоживання. Висока інтенсивність надходження інформації вимагає пошуку нових архітектурних та алгоритмічних рішень, оскільки традиційні обчислювальні системи не здатні забезпечити необхідну швидкодію.

Аналіз існуючих методів проєктування засобів для мобільних робототехнічних систем показує, що значна частина рішень не орієнтована на роботу в режимі реального часу та не забезпечує високу ефективність використання апаратних ресурсів. Використання виключно універсальних мікропроцесорів із розробкою спеціалізованого програмного забезпечення є доступним, проте характеризується невисокою швидкодією, а також функціональною і структурною надлишковістю. Для забезпечення режиму реального часу та екстремального підвищення продуктивності необхідно впроваджувати апаратну спеціалізацію, конвеєризацію, просторовий паралелізм обчислень та чітке узгодження інтенсивності надходження даних з обчислювальною здатністю системи.

Постановка задачі. Об'єкт дослідження – процеси опрацювання даних у реальному часі інтелектуальними (нейрокомп'ютерними) засобами мобільних робототехнічних платформ. Предмет дослідження – методи та алгоритми узгоджено-паралельної обробки даних, їх просторово-часового відображення у структури апаратних засобів Мета роботи – підвищити швидкодію інтелектуальних засобів мобільних робототехнічних платформ шляхом розроблення методів відображення алгоритмів в узгоджено-паралельні структури та забезпечення узгодженості інтенсивності надходження потоків даних з обчислювальною здатністю апаратних засобів.

Основний матеріал. Підвищення швидкодії інтелектуальних засобів досягається шляхом апаратного відображення структури алгоритму розв'язання задачі безпосередньо у спеціалізовану архітектуру. Виходячи з вимог робототехніки, найперспективнішими є два варіанти апаратної реалізації.

1. Гібридна архітектура: Використання універсального обчислювального ядра, доповненого спеціалізованими НВІС-пристроями (надвеликими інтегральними схемами). Ці пристрої беруть на себе реалізацію найбільш часомістких алгоритмів функціонування штучних нейронних мереж (ШНМ). Таке поєднання гарантує швидку обробку нерегулярних алгоритмів із великою кількістю логічних операцій.

2. Спеціалізована алгоритмічна НВІС-система: Архітектура, яка повністю апаратно відображає структуру інтелектуального алгоритму. Цей варіант орієнтований на обробку найбільш інтенсивних потоків даних у реальному часі. Висока швидкодія тут досягається за рахунок відсутності проміжних пересилок інформації під час обчислень та суттєвого спрощення функцій місцевого управління.

Алгоритмічна оптимізація та просторово-часове відображення. Ефективність апаратного прискорення безпосередньо залежить від адаптації самих алгоритмів. Для НВІС-реалізації алгоритми повинні бути добре структурованими, мати детерміноване переміщення даних та орієнтуватися на множину взаємозв'язаних процесорних елементів (ПЕ). В першу чергу розробляються алгоритми з рекурсивною та локальною залежністю. Локалізація дозволяє

здійснювати обміни даними лише між найближчими сусідніми ПЕ, що усуває затримки, характерні для глобальних пересилок.

Процес підвищення швидкодії включає декомпозицію алгоритму на функціональні оператори та проєктування комунікацій між ними. Метою є створення потокового графу, де функціональні оператори розбиваються на прості операції, що мають приблизно однаковий час виконання.

Конвеєризація та узгодження інформаційних потоків. Основним методом радикального підвищення швидкодії є просторово-часове відображення алгоритмів у узгоджено-паралельні структури. Для обробки потоків даних у реальному часі використовуються синхронні структури з конвеєрною реалізацією, в яких відбувається суміщення в часі виконання різних операцій над різними даними.

Конвеєризація передбачає поділ системи на сходинки з використанням буферної пам'яті. Для максимізації швидкодії оператори у кожній сходинці мають бути простими. Ключовим фактором швидкодії є узгодження інтенсивності надходження даних із обчислювальною здатністю системи. Обчислювальна здатність конвеєра залежить від такту роботи апаратних засобів, який, у свою чергу, визначається часом звертання до пам'яті та часом обчислення найскладнішого оператора.

Для досягнення високої ефективності та швидкості застосовуються такі методи балансування: зміна конвеєрного такту (мінімізація часу виконання на найскладнішому етапі конвеєра); масштабування каналів (зміна кількості та розрядності каналів надходження даних в операційних пристроях в залежності від вимог потоку) та укрупнення операторів (якщо пропускна здатність перевищує інтенсивність потоку, функціональні оператори або яруси конвеєра об'єднуються, що дозволяє оптимізувати апаратні витрати та зберегти високу швидкість).

Мінімізація апаратних витрат як інструмент прискорення. Підвищення швидкодії неможливе без врахування апаратних витрат, оскільки надлишковість уповільнює роботу системи через ускладнення маршрутизації сигналів на кристалі. Основними шляхами оптимізації, які ведуть до зростання загальної швидкодії системи, є вибір ефективних методів та алгоритмів для НВІС-реалізації; зменшення розрядності операційних пристроїв і пам’яті до мінімально необхідного для задачі рівня, що скорочує затримки на логічних вентилях; скорочення кількості та розрядності каналів передачі даних.

Висновки. Підвищення швидкодії інтелектуальних засобів мобільних робототехнічних платформ вимагає відходу від використання виключно універсальних мікропроцесорів. Найбільш ефективним підходом є впровадження спеціалізованих НВІС-систем або гібридних архітектур, що базуються на принципах конвеєризації та просторового паралелізму. Ключовим математичним і структурним інструментом для забезпечення обробки у реальному часі є просторово-часове відображення алгоритмів та узгодження обчислювальної здатності обладнання з інтенсивністю інформаційних потоків. Застосування локальних алгоритмів обробки, рівномірне навантаження на сходинки конвеєра та відмова від проміжних пересилок даних дозволяють досягти необхідної продуктивності для сучасних мобільних роботів при збереженні жорстких обмежень щодо габаритів та енергоспоживання.

Список літератури

1. Kang, S. K., Kang, S., & Yoo, H. J. (2019). A study on FPGA acceleration for deep neural networks using dynamic precision scaling. IEEE Transactions on Circuits and Systems for Video Technology, 29(6), 1539-1549. https://doi.org/10.1109/TCSVT.2018.2845390

2. Tsmots, I., Teslyuk, V., Łukaszewicz, A., Lukashchuk, Y., Kazymyra, I., Holovatyy, A., & Opotyak, Y. (2023). An approach to the implementation of a neural network for cryptographic protection of data transmission at UAV. *Drones*, 7(507). https://doi.org/10.3390/drones7080507

3. Tsmots, I., Teslyuk, V., Kryvinska, N., Skorokhoda, O., Kazymyra, I. Development of a generalized model for parallel-streaming neural element and structures for scalar product calculation devices. Journal of Supercomputing, 2023, 79(5), pp. 4820–4846.

Цмоць І.Г.[1], Петрина В.В.[2]
[1]доктор технічних наук, професор, кафедра автоматизованих систем управління, Національний університет «Львівська політехніка»
[2]аспірант, кафедра автоматизованих систем управління, Національний університет «Львівська політехніка»
Науковий керівник: д.т.н., проф. Цмоць І.Г., кафедра автоматизованих систем управління, Національний університет «Львівська політехніка»

ІНФОРМАЦІЙНІ ПОТОКИ, ЗАСОБИ ЗБОРУ ТА ІНТЕГРАЦІЇ ДАНИХ У СМАРТ-ПІДПРИЄМСТВАХ

Актуальність. Сучасний розвиток концепції Індустрії 4.0, Інтернету речей (IoT), кіберфізичних систем, штучного інтелекту та Big Data зумовлює активну цифрову трансформацію промислових підприємств і формування смарт-підприємств нового покоління [1]. У таких умовах ефективність функціонування виробничих, логістичних, управлінських та аналітичних процесів значною мірою залежить від організації інформаційних потоків, засобів збору даних, а також систем інтеграції та злиття інформації з різнорідних джерел.

Актуальність дослідження обумовлена тим, що сучасні смарт-підприємства функціонують у середовищі безперервного генерування великих обсягів даних, які надходять від сенсорних систем, IoT-пристроїв, виробничого обладнання, ERP-, MES- та SCADA-систем, хмарних сервісів і цифрових платформ. Відсутність ефективних механізмів інтеграції та аналітичної обробки таких даних призводить до зниження достовірності інформації, збільшення затримок у прийнятті управлінських рішень та зменшення ефективності цифрового виробництва [2].

Постановка задачі. Об'єктом дослідження є інформаційні потоки, засоби збору, інтеграції та злиття даних у смарт-підприємствах. Предметом дослідження є математичні моделі інформаційних потоків, засобів збору даних, а також процесів інтеграції та злиття даних у смарт-підприємствах. Метою дослідження є підвищення ефективності роботи смарт-підприємств шляхом розроблення математичних моделей інформаційних потоків, засобів збору, інтеграції та злиття даних, які забезпечують автоматизацію процесів обробки інформації, підвищення достовірності аналітики та підтримку інтелектуального управління цифровим виробництвом в умовах Індустрії 4.0

Основний матеріал. Смарт-підприємство являє собою цифрово інтегровану систему, у якій виробничі, логістичні, управлінські та аналітичні процеси функціонують на основі постійного обміну даними між обладнанням, інформаційними системами та персоналом. Таке смарт-підприємство можна розглядати як інформаційну систему, що описується множиною:

$$SE = \{D, P, C, A, U\}\ , \tag{1}$$

де D — множина джерел даних; P — множина інформаційних потоків; C — комунікаційна інфраструктура; A — аналітичні модулі; U — система управління.

Основу функціонування такого підприємства формують інформаційні потоки та засоби збору даних, що забезпечують отримання, передавання, обробку та аналіз інформації у режимі реального часу.

Класифікація інформаційних потоків.

Усі інформаційні потоки у смарт-підприємстві можна записати у вигляді множини:

$$P = \{p_1, \ldots, p_i, \ldots p_n\}, \tag{2}$$

де p_i — окремий i-й інформаційний потік.

Кожний i-й інформаційний потік описується кортежем:

$$p_i = \{s_i, r_i, t_i, q_i, a_i\}, \tag{3}$$

де s_i – джерело інформації, r_i — отримувач інформації, t_i - часові характеристики передачі, q_i — обсяг або інтенсивність даних, a_i — рівень автоматизації.

Інформаційні потоки у смарт-підприємстві можна класифікувати за різними ознаками залежно від їх призначення, структури та способу функціонування.

Основними ознаками класифікації є: напрямок руху інформації, функціональне призначення, швидкість передачі та рівень автоматизації.

За напрямком руху інформаційні потоки поділяються на три групи: вхідні, внутрішні та вихідні і записуються так:

$$P = P_{in} \cup Pout_{int}, \quad (4)$$

де P_{in} — множина вхідних потоків, P_{int} — множина внутрішніх потоків, P_{out} — множина вихідних потоків.

За функціональним призначенням інформаційні потоки у смарт-підприємстві доцільно поділяти на п'ять груп: управлінські, виробничі, логістичні, фінансові та інноваційно-аналітичні потоки і записуються так:

$$P = P_m \cup P_p \cup P_l \cup P_f \cup P_a, \quad (5)$$

де P_m — управлінські потоки, Pp — виробничі потоки, P_l — логістичні потоки, P_f — фінансові потоки, P_a — інноваційно-аналітичні потоки.

$$P = P_{rt} \cup P_{per} \cup P_{arch}, \quad (6)$$

де P_{rt} — потоки реального часу, P_{per} — періодичні потоки, P_{arch} — архівні потоки.

Швидкість передачі визначається функцією:

$$v(p_i) = \frac{q_i}{\Delta t_i}, \quad (7)$$

де q_i — обсяг переданих даних, Δt_i — час передачі.

За рівнем автоматизації інформаційні потоки у смарт-підприємстві діляться: на автоматизовані, напівавтоматизовані та ручні інформаційні потоки і записуються так:

$$P = P_{auto} \cup P_{semi} \cup P_{man}, \quad (8)$$

де P_{auto} — автоматизовані потоки, P_{semi} — напівавтоматизовані потоки, P_{man} — ручні потоки.
Рівень автоматизації визначимо коефіцієнтом:

$$\alpha_i \in [0,1], \quad (9)$$

де α_i=1 — повністю автоматизований потік, α_i=0 — ручний потік, 0,3< α_i<0,8 – напівавтоматизований потік.

Таким чином, класифікація інформаційних потоків у смарт-підприємстві за напрямком руху, функціональним призначенням, швидкістю передачі та рівнем автоматизації забезпечує систематизацію процесів збору, передачі й обробки даних та створює основу для ефективного функціонування цифрового виробництва в умовах Індустрії 4.0.

Засоби збору даних у смарт-підприємствах.

Математично систему засобів збору даних у смарт-підприємстві можна подати у вигляді такої множини:

$$DS = \{S, I, N, P, A\}, \quad (10)$$

де S— множина сенсорних засобів, I — множина ідентифікаційних засобів, N — множина мережевих та телекомунікаційних засобів, P — множина IoT-платформ, A — множина аналітичних засобів обробки даних.

У смарт-підприємствах одними із головних засобів збору є сенсори S, які діляться на три групи: фізичні, біометричні та оптичні [3].

Аналітичні засоби A у смарт-підприємствах забезпечують обробку, аналіз, прогнозування та інтелектуальне використання даних, що надходять із сенсорних систем, IoT-пристроїв, виробничого обладнання, ERP-, MES- та SCADA-систем. Вони формують основу інтелектуального управління цифровим виробництвом та підтримують прийняття управлінських рішень у режимі реального часу [4].

Математично множину аналітичних засобів можна подати так:

$$A = \{A_{BD}, A_{AI}, A_{ML}, A_{BI}, A_{DT}\}, \quad (11)$$

де A_{BD} — засоби Big Data-аналітики, A_{AI} — системи штучного інтелекту, A_{ML} — засоби машинного навчання, A_{BI} — системи бізнес-аналітики (Business Intelligence), A_{DT} — системи цифрових двійників (Digital Twins).

Розроблення засобів інтеграції та злиття даних у смарт-підприємстві.

Засоби інтеграції та злиття даних [5] доцільно описати множиною:

$$IF = \{M, API, PR, ETL, DF\}, \quad (12)$$

де M — middleware-платформи, API — програмні інтерфейси обміну даними, PR — промислові протоколи, ETL — засоби вилучення, перетворення та завантаження даних, DF — засоби злиття даних.

Отриманий результат інтеграції та злиття даних використовується аналітичними модулями та системою управління:

$$U(t) = F_u(Y(t), A), \tag{13}$$

де $U(t)$— управлінський вплив у момент часу t, F_u — функція формування управлінського рішення, A — аналітичні модулі смарт-підприємства.

Отже, повна модель інтеграції та злиття даних у смарт-підприємстві має вигляд:

$$D \to X(t) \to Ffusion_{int}, \tag{14}$$

де D — множина джерел даних, $X(t)$ — множина вхідних потоків даних, F_{int} — функція інтеграції даних, $Z(t)$ — інтегровані дані, F_{fusion} — функція злиття даних, $Y(t)$ — результуюча модель даних, A — аналітичні модулі, $U(t)$ — система управління.

Таким чином, математичний опис засобів інтеграції та злиття даних дозволяє формалізувати процес об'єднання різнорідної інформації, підвищити її достовірність, зменшити вплив похибок сенсорів і забезпечити формування обґрунтованих управлінських рішень у смарт-підприємстві.

Висновки.

1. Розроблено математичний опис інформаційних потоків смарт-підприємства та виконано їх класифікацію за напрямком руху, функціональним призначенням, швидкістю передачі й рівнем автоматизації, що дозволяє формалізувати процеси обміну та обробки даних у цифровому виробництві.

2. Досліджено та математично описано засоби збору даних у смарт-підприємствах, зокрема сенсорні, ідентифікаційні, мережеві та аналітичні засоби, а також IoT-платформи, які забезпечують інтеграцію фізичних об'єктів із цифровою інфраструктурою підприємства.

3. Розроблено математичні моделі інтеграції та злиття даних, які забезпечують об'єднання інформації з різнорідних джерел, підвищення достовірності аналітичної обробки даних та підтримку інтелектуального управління смарт-підприємством в умовах Індустрії 4.0.

Список літератури

1. Teslyuk V., Tsmots I., Nazarkevych H. Methods and instruments of adaptive control over an intelligent enterprise using the weak signals // International Journal of Services Technology and Management. 2025. Vol. 30, iss. 4. P. 284–305.

2. Zhao Y. Development of big data assisted enterprise resource planning framework. PLOS ONE. 2024.

3. Sahal R., Breslin J. G., Ali M. I. Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. Journal of Manufacturing Systems. 2022. Vol. 54. P. 138–151. DOI: 10.1016/j.jmsy.2019.11.004.

4. Lu Y., Liu C., Wang K. I.-K., Huang H., Xu X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robotics and Computer-Integrated Manufacturing. 2021. Vol. 61. P. 101837. DOI: 10.1016/j.rcim.2019.101837.

5. Qi Q., Tao F. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. IEEE Access. 2021. Vol. 6. P. 3585–3593. DOI: 10.1109/ACCESS.2018.2793265.

Вдодович Ю., Вівчар В.
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Батько Ю.М., кафедра КІ ЗУНУ

АНАЛІЗ СКЛАДНИХ ДИНАМІЧНИХ СТРУКТУР ДАНИХ У МОВІ ПРОГРАМУВАННЯ С++

Вступ. У сучасній інформатиці динамічні структури даних займають ключове місце, оскільки вони забезпечують гнучкість і ефективність роботи програмного коду. Мова програмування С++ надає широкі можливості для реалізації таких структур, що дозволяє створювати високопродуктивні системи, здатні адаптуватися до змінних умов обробки інформації. Використання динамічних структур даних у С++ обґрунтоване потребою в оптимальному управлінні пам'яттю, можливістю масштабування та забезпеченням високої швидкості виконання алгоритмів. Саме ці характеристики роблять їх незамінними у процесі розробки програмного забезпечення, орієнтованого на складні обчислювальні завдання [1].

Постановка задачі. Об'єкт дослідження – підходи зберігання та обробки великих об'ємів даних. Предмет дослідження – складні динамічні структури даних в мові програмування С++. Головна мета даного дослідження полягає аналізі та описі динамічних структур для зберігання та обробки даних в мові С++, виділення їх основних характеристик та особливостей використання.

Основний матеріал. Динамічні структури даних у С++ дозволяють розробникам працювати з інформацією, обсяг якої не є відомим заздалегідь. Це означає, що вони здатні змінювати свій розмір під час виконання програми, що особливо важливо у випадках, коли дані надходять у реальному часі або їхня кількість залежить від зовнішніх факторів. У цьому контексті використання статичних структур, таких як масиви з фіксованим розміром, є обмеженим, оскільки вони не можуть адаптуватися до змінних умов. Динамічні структури, навпаки, забезпечують необхідну гнучкість і дозволяють ефективно управляти ресурсами.

Особливості використання кожної динамічної структури даних у С++ визначаються їхньою внутрішньою організацією та призначенням. Наприклад, список є структурою, що складається з вузлів, кожен з яких містить елемент даних і посилання на наступний або попередній вузол. Це забезпечує можливість швидкого додавання та видалення елементів у будь-якому місці структури. Проте доступ до конкретного елемента потребує послідовного проходження, що збільшує часові затримки під час обробки даних. Вектор, навпаки, є динамічним масивом, який дозволяє здійснювати швидкий доступ до елементів за індексом, але операції вставки та видалення можуть бути менш ефективними через необхідність зміщення інших елементів. Черга та стек реалізують специфічні принципи організації даних. Черги опрацьовують дані за схемою «першим прийшов – першим обробляєшся», а стек за принципом «останнім прийшов – першим оброблений». Це робить їх корисними для моделювання процесів, де порядок обробки має вирішальне значення [2].

Важливою особливістю використання динамічних структур у С++ є можливість управління пам'яттю на низькому рівні. Мова надає програмісту інструменти для виділення та звільнення пам'яті вручну, що дозволяє оптимізувати роботу програми відповідно до конкретних вимог. Це створює умови для ефективного використання ресурсів, але водночас накладає відповідальність за правильність реалізації. Оскільки, помилки у керуванні пам'яттю можуть призвести до серйозних проблем. У цьому контексті використання стандартної бібліотеки шаблонів (STL) є доцільним, оскільки вона забезпечує готові реалізації динамічних структур із вбудованими механізмами управління пам'яттю [3].

Перевагами використання динамічних структур даних у програмному коді обґрунтовується кількома ключовими факторами. По-перше, вони забезпечують адаптивність системи до змінних умов, що є критично важливим у сучасних інформаційних технологіях. По-друге, динамічні структури дозволяють ефективно працювати з великими об'ємами даних, що робить їх незамінними у сфері аналізу інформації, машинного навчання та обробки сигналів. По-третє, вони сприяють підвищенню продуктивності програмного забезпечення, оскільки

дозволяють оптимізувати виконання алгоритмів шляхом вибору найбільш відповідної структури для конкретного завдання.

В процесі аналізу динамічних структур здійснено їх порівняння на основі трьох характеристик: швидкість доступу до елементів, ефективність операцій вставки та видалення, а також використання пам'яті. Вибір саме цих характеристик обґрунтований тим, що вони визначають практичну цінність структури у реальних умовах програмування. Швидкість доступу є критичною для завдань, де необхідно швидко отримувати інформацію. Ефективність операцій вставки та видалення визначає гнучкість структури у випадках, коли дані постійно змінюються. Використання пам'яті є важливим фактором у контексті оптимізації ресурсів, особливо при роботі з великими обсягами інформації. Таким чином, аналіз на основі цих характеристик дозволяє зробити обґрунтований вибір структури для конкретного завдання

У таблиці 1 наведено порівняльний аналіз динамічних структур даних у C++.

Таблиця 1 - Порівняльна таблиця динамічних структур даних у C++

Структура	Швидкість доступу	Вставка/видалення	Використання пам'яті
Вектор	Висока при доступі за індексом	Середня, потребує зміщення елементів	Ефективне, але може бути надлишковим при розширенні
Список	Низька, потребує послідовного проходження	Висока, швидке додавання та видалення	Витрати на зберігання посилань
Стек	Висока для верхнього елемента	Висока, операції виконуються миттєво	Ефективне, мінімальні витрати
Черга	Висока для першого елемента	Висока, швидке додавання та видалення	Ефективне, мінімальні витрати
Бінарні дерева	Низька, потребує послідовного проходження	Висока, швидке додавання та видалення	Витрати на зберігання посилань

Висновки. Проведене дослідження проілюструвало, що динамічні структури даних у мові програмування C++ є фундаментальним інструментом для створення ефективного та гнучкого програмного забезпечення. Їх використання забезпечує адаптивність системи, оптимізацію ресурсів та підвищення продуктивності. Особливості кожної структури визначають її практичну цінність у конкретних умовах, а порівняння на основі ключових характеристик дозволяє зробити обґрунтований вибір. Таким чином, динамічні структури даних у C++ є не лише теоретично важливими, але й практично незамінними у сучасній інформатиці.

Список літератури

1. В. Григолович, Алгоритмізація та програмування. Частина 4: Динамічні та рекурсивні структури даних, Магнолія, Львів, Україна, 2026.

2. Є.О. Іванов, Я.М. Ліндер, К.А. Жереб, О.О. Супрун, Представлення та обробка динамічних структур даних, Київський національний університет імені Тараса Шевченка, Київ, Україна, 2025.

3. J. Smith and R. Johnson, Advanced Algorithms and Data Structures in C++20, Springer, Cham, Switzerland, 2025.

Ковбаса Д., Нукало В.
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Батько Ю.М., кафедра КІ ЗУНУ

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ IOT-ПРИСТРОЇВ НА БАЗІ RASPBERRY PI PICO W ТА МОВИ ПРОГРАМУВАННЯ MICROPYTHON

Вступ. Сучасна парадигма розвитку інформаційних технологій визначається стрімким поширенням концепції Інтернету речей (IoT), яка передбачає інтеграцію фізичних пристроїв у глобальні мережі для збору, обробки та передачі даних. У цьому контексті особливого значення набуває використання мікроконтролерів, здатних забезпечити ефективну взаємодію між сенсорами, виконавчими механізмами та хмарними сервісами. Одним із найбільш перспективних рішень є Raspberry Pi Pico W, що поєднує компактність, низьке енергоспоживання та підтримку бездротових технологій. У свою чергу, MicroPython виступає як оптимальна мова програмування для реалізації IoT-проєктів завдяки простоті синтаксису, широким можливостям інтеграції та підтримці апаратних інтерфейсів [1].

Постановка задачі. Об'єкт дослідження – технології проєктування та впровадження елементів інтелектуалізованих систем. Предмет дослідження – IoT-пристрої на базі мікроконтролера Raspberry Pi Pico W. Головна мета даного дослідження полягає дослідженні технічних характеристик та можливостей реалізації IoT пристроїв з використання мови програмування MicroPython.

Основний матеріал. MicroPython є інтерпретованою мовою програмування, спеціально адаптованою для роботи на мікроконтролерах. Її синтаксис базується на Python, що робить її зрозумілою та доступною навіть для початківців. Важливою перевагою є можливість швидкої розробки та тестування коду без необхідності компіляції, що значно скорочує час реалізації проєктів. Це особливо важливо у випадках, коли необхідно перевірити роботу сенсора або налаштувати параметри зв'язку. MicroPython має модульну структуру, що дозволяє використовувати лише ті бібліотеки, які потрібні для конкретного проєкту, тим самим оптимізуючи використання пам'яті. Вона також підтримує роботу з протоколами MQTT та HTTP, що забезпечує інтеграцію з хмарними сервісами та іншими IoT-платформами. MicroPython має вбудовані бібліотеки для роботи з апаратними інтерфейсами, що дозволяє легко керувати сенсорами, дисплеями та іншими компонентами. Завдяки цьому MicroPython дозволяє створювати комплексні системи, які можуть взаємодіяти з різними елементами цифрової інфраструктури [2].

Raspberry Pi Pico W є мікроконтролером на базі чипа RP2040, який має двоядерний процесор ARM Cortex-M0+ та підтримку Wi-Fi. Його ключова перевага полягає у поєднанні високої продуктивності з низькою вартістю, що робить його доступним для широкого кола дослідників та розробників. Raspberry Pi Pico W має низку особливостей, які визначають його практичну цінність у контексті IoT. По-перше, він підтримує роботу з низьковольтними сенсорами, що дозволяє створювати енергоефективні системи. По-друге, його архітектура забезпечує можливість паралельної обробки даних завдяки двоядерному процесору, що підвищує продуктивність. По-третє, інтегрований модуль Wi-Fi дозволяє реалізувати бездротову передачу даних до хмарних сервісів або мобільних додатків. Це відкриває можливості для створення систем моніторингу, автоматизації та управління у реальному часі. Крім того, Pico W має компактні розміри, що дозволяє інтегрувати його у пристрої з обмеженим простором. Це забезпечує компактність конструкції та знижує витрати на реалізацію проєкту. Крім того, Pico W має широкий набір інтерфейсів, включаючи GPIO, I2C, SPI та UART, що дозволяє інтегрувати його з різноманітними сенсорами та виконавчими пристроями [3].

Для об'єктивного аналізу доцільно порівняти Raspberry Pi Pico W з двома популярними альтернативами ESP32 та Arduino Nano IoT. Основними критеріями вибору виступають продуктивність, мережеві можливості, енергоспоживання та вартість. Продуктивність визначає здатність мікроконтролера виконувати складні обчислення та обробляти великі обсяги даних у реальному часі. Це критично для систем, що працюють із сенсорними мережами або

алгоритмами машинного навчання. Мережеві можливості є ключовими для IoT, адже пристрої повинні взаємодіяти між собою та з хмарними сервісами. Наявність Wi-Fi та Bluetooth розширює спектр застосувань. Енергоспоживання визначає тривалість автономної роботи пристрою, що особливо важливо для мобільних або портативних систем. Вартість є практичним фактором, який впливає на економічну доцільність використання мікроконтролера у масових проєктах чи навчальних цілях. Саме ці характеристики визначають практичну придатність мікроконтролера у проєктах IoT, оскільки вони безпосередньо впливають на швидкість обробки даних, можливість інтеграції у мережу, тривалість автономної роботи та економічну ефективність.

У таблиці 1 наведено порівняльний аналіз мікроконтролерів для реалізації IoT пристроїв.

Таблиця 1 - Порівняльна таблиця мікроконтролерів для реалізації IoT пристроїв

Критерій	Raspberry Pi Pico W	ESP32	Arduino Nano IoT
Продуктивність	ARM Cortex-M0+, 133 MHz, 264 KB RAM	Xtensa LX6, до 240 MHz, 520 KB RAM	ARM Cortex-M0+, 48 MHz, 32 KB RAM
Мережеві можливості	Wi-Fi (802.11n)	Wi-Fi + Bluetooth (Classic + BLE)	Wi-Fi (802.11b/g/n)
Енергоспоживання	Дуже низьке, оптимізоване для сенсорних систем	Вищий рівень через потужний процесор	Середнє, оптимізоване для простих IoT
Вартість	~6–7 USD	~8–10 USD	~18–20 USD

Одним із прикладів є створення системи моніторингу температури та вологості у приміщенні. Raspberry Pi Pico W підключається до сенсора, а MicroPython використовується для написання програми, яка зчитує дані та передає їх через Wi-Fi до хмарного сервісу. Це дозволяє користувачеві отримувати інформацію у реальному часі та налаштовувати автоматичні сповіщення. Іншим прикладом є система «розумного освітлення», де Pico W керує світлодіодними лампами на основі даних від датчика руху. Програма на MicroPython забезпечує логіку роботи системи, включаючи автоматичне вимкнення світла при відсутності руху. Ще одним прикладом є створення портативного пристрою для моніторингу якості повітря, який збирає дані з сенсора CO2 та передає їх на мобільний додаток. Усі ці приклади демонструють практичну цінність використання Pico W та MicroPython у реальних умовах.

Висновки. Проєктування та розробка IoT-пристроїв на базі Raspberry Pi Pico W та мови програмування MicroPython є перспективним напрямом сучасної інформатики. Поєднання цих елементів обґрунтований їхньою функціональністю, доступністю та взаємною сумісністю. Raspberry Pi Pico W забезпечує апаратну основу з високою продуктивністю та підтримкою бездротових технологій, тоді як MicroPython надає простий та ефективний інструмент для програмування. Разом вони створюють середовище, яке дозволяє швидко та ефективно реалізовувати IoT-рішення.

Список літератури

1. H. Nguyen, *Embedded Systems with MicroPython: Real-Time IoT Applications*, CRC Press, Boca Raton, FL, USA, 2025.

2. K. Patel, *IoT Development with Raspberry Pi Pico W and MicroPython*, Apress, New York, USA, 2024.

3. S. Gupta, *Secure IoT Architectures with MicroPython and Raspberry Pi Pico W*, Springer, Berlin, Germany, 2025.

Слюсар М., Франчишин Ю.
студенти 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Батько Ю.М., кафедра КІ ЗУНУ

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ IOT-ПРИСТРОЇВ НА БАЗІ RASPBERRY PI PICO W ТА МОВИ ПРОГРАМУВАННЯ MICROPYTHON

Вступ. У сучасній інформатиці та комп'ютерному зорі завдання обробки та розпізнавання текстів на цифрових зображеннях займає ключове місце. Ця проблема має практичне значення у багатьох сферах від автоматизації документообігу та цифровізації архівів до створення систем штучного інтелекту, здатних інтерпретувати візуальну інформацію. Алгоритми, що застосовуються для розпізнавання текстів, поєднують методи обробки зображень, машинного навчання та лінгвістичного аналізу. Їхня ефективність визначається здатністю працювати з різними типами даних, включаючи рукописні тексти, друковані документи та зображення низької якості [1].

Постановка задачі. Об'єкт дослідження – методи та алгоритми обробки цифрових кольорових зображень. Предмет дослідження – алгоритми розпізнавання текстової інформації на цифрових зображеннях. Головна мета даного дослідження полягає в цілісному дослідження усіх етапів обробки та розпізнавання цифрових зображень. Що в подальшому дозволить підібрати оптимальний набір алгоритмів для здійснення попередньої обробки, аналізу та розпізнавання текстової інформації а цифрових зображеннях.

Основний матеріал. Процес розпізнавання текстів на цифрових зображеннях складається з кількох послідовних етапів, кожен з яких має власну специфіку та визначає якість кінцевого результату. Першим етапом є попередня обробка зображення, яка включає нормалізацію яскравості, контрасту та шумозниження. Ці операції дозволяють підготувати зображення до подальшого аналізу, зменшити вплив артефактів та підвищити чіткість текстових елементів. Важливим аспектом є також бінаризація, яка перетворює зображення у двоколірний формат, що значно спрощує виділення символів.

Другим етапом є сегментація, яка полягає у виділенні текстових областей на зображенні. Сегментація може здійснюватися на рівні рядків, слів або окремих символів. Для цього застосовуються алгоритми пошуку контурів, аналізу проекцій та кластеризації. Ефективність сегментації визначає точність подальшого розпізнавання, оскільки неправильне виділення символів може призвести до помилок у результатах. Особливу складність становить сегментація рукописних текстів, де символи можуть накладатися один на одного або мати нерівномірні інтервали.

Наступним етапом є отримання ознак, що передбачає перетворення виділених символів у набір характеристик, що описують їхню форму та структуру. Ознаки можуть включати геометричні параметри, такі як висота, ширина та співвідношення сторін, а також більш складні характеристики, наприклад, гістограми градієнтів або хвильові перетворення. Екстракція ознак є критично важливою для забезпечення стійкості алгоритму до варіацій у написанні символів та умовах зображення.

Після етапу виділення характерних ознак проводиться етап класифікації, який полягає у віднесенні символів до певних класів на основі їхніх ознак. Для цього застосовуються методи машинного навчання, включаючи нейронні мережі, метод опорних векторів та байєсівські класифікатори. Сучасні системи розпізнавання текстів часто використовують глибокі нейронні мережі, які здатні автоматично навчатися ознакам та забезпечувати високу точність навіть у складних умовах. Класифікація завершується формуванням текстового виходу, який може бути представленим у вигляді рядка символів або структурованого документа.

Останнім етапом є постобробка, яка включає корекцію помилок, перевірку орфографії та синтаксичний аналіз. Постобробка дозволяє підвищити якість результатів, особливо у випадках, коли алгоритм класифікації допустив помилки. Для цього використовуються словники, граматичні правила та статистичні моделі мови. Постобробка також може включати інтеграцію з іншими системами, наприклад, базами даних або пошуковими сервісами [2].

Алгоритми обробки та розпізнавання текстів мають низку особливостей, які визначають їхню практичну цінність. По-перше, вони повинні бути стійкими до варіацій у якості зображень, включаючи різні рівні шуму, освітлення та роздільної здатності. Окрім того, алгоритми повинні враховувати різноманітність шрифтів та стилів написання, що особливо актуально для рукописних текстів. Важливим є те, що вони повинні забезпечувати високу швидкість обробки, що є критично важливим для систем реального часу. Нарешті, алгоритми повинні бути здатними працювати з багатомовними текстами, що розширює їхню сферу застосування.

На практиці алгоритми обробки та розпізнавання текстів включають системи оптичного розпізнавання символів (OCR), які застосовуються для цифровізації друкованих документів. Іншим прикладом є системи автоматичного перекладу, які використовують розпізнавання текстів для обробки зображень із написами. Системи оптичного розпізнавання символів є комплексом алгоритмів та програмних рішень, призначених для автоматичного перетворення текстової інформації, представленої у вигляді цифрового зображення, у машинно-читаний формат. Їхня основна мета полягає у забезпеченні можливості подальшої обробки, пошуку та аналізу текстових даних без необхідності ручного введення. OCR-технології стали фундаментом для цифровізації друкованих документів, архівів та рукописних матеріалів, а також для створення інтелектуальних систем, здатних інтерпретувати візуальну інформацію [3].

Сучасні системи OCR використовують глибокі нейронні мережі, що дозволяє досягати високої точності навіть у випадках з низькою якістю зображень або рукописними текстами. Вони здатні працювати з багатомовними документами, підтримують різні шрифти та стилі написання. Важливою особливістю є інтеграція з хмарними сервісами, що забезпечує масштабованість та можливість обробки великих обсягів даних у реальному часі. Системи OCR також активно застосовують методи контекстного аналізу, які дозволяють враховувати семантику тексту та зменшувати кількість помилок.

OCR використовується у сфері документообігу для автоматичного введення даних із друкованих форм у електронні системи. У бібліотеках та архівах ці технології застосовуються для цифровізації книжкових фондів. У транспортній галузі OCR використовується для розпізнавання номерних знаків автомобілів. У фінансовій сфері системи OCR інтегруються у мобільні додатки для автоматичного зчитування реквізитів із платіжних документів. У медицині вони дозволяють обробляти рукописні записи лікарів та інтегрувати їх у електронні медичні картки.

Висновки. Проведений аналіз алгоритмів обробки та розпізнавання текстів на цифрових зображеннях проілюстрував, що цей процес є складним та складається з багатьох етапів. Кожен із цих етапів має ключове значення для забезпечення високої точності та практичної цінності результатів. Алгоритми повинні бути стійкими до варіацій у якості зображень, різноманітності шрифтів та умов освітлення. Їхнє практичне застосування охоплює широкий спектр сфер, включаючи документообіг, безпеку, медицину та переклад. Таким чином, розвиток алгоритмів обробки та розпізнавання текстів є стратегічно важливим для сучасної інформатики та має значний потенціал для подальших досліджень.

Список літератури

1. О. Коваль, *Цифрова обробка та аналіз зображень: сучасні алгоритми та застосування*, Видавництво «Львівська політехніка», Львів, Україна, 2024.

2. F. Cuevas, P. L. Mazzeo, A. Bruno, *Digital Image Processing – Latest Advances and Applications*, BoD – Books on Demand, Germany, July 2024, 234 p.

3. K. C. Santosh, A. Makkar, M. Conway, A. K. Singh, A. Vacavant, A. Abou el Kalam, M.-R. Bouguelia, R. Hegadi (eds.), *Recent Trends in Image Processing and Pattern Recognition: 6th International Conference RTIP2R 2023, Derby, UK, Revised Selected Papers*, Springer Nature, January 2024, 413 p.

Григорів М.-В.В.
студентка 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Піцун О.Й., кафедра КІ ЗУНУ

РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ОНЛАЙН-БІБЛІОТЕКИ З ІНТЕГРАЦІЄЮ ШТУЧНОГО ІНТЕЛЕКТУ

Вступ. Цифровізація бібліотечних послуг є одним із ключових напрямів розвитку освітнього та культурного середовища. Зростання кількості електронних публікацій та читачів, які надають перевагу онлайн-ресурсам, зумовлює потребу у сучасних веб-платформах для роботи з книжковими фондами. За даними Міжнародної федерації бібліотечних асоціацій (IFLA), понад 70 % бібліотек у країнах ЄС впровадили цифрові каталоги, проте лише незначна їх частина пропонує персоналізовані сервіси на основі штучного інтелекту [1].

Вітчизняні бібліотечні веб-ресурси здебільшого обмежуються статичним каталогом без інтеграції зовнішніх книжкових баз та інтелектуальних рекомендаційних систем [2]. Алгоритми персоналізації суттєво підвищують залученість користувачів і час їх взаємодії з платформою [3]. Отже, розробка інтегрованої бібліотечної платформи з підтримкою AI-рекомендацій, рольового розмежування доступу та агрегацією даних із зовнішніх каталогів є актуальним завданням, що поєднує сучасні вебтехнології та методи обробки природної мови.

Постановка задачі. Об'єктом дослідження є процес автоматизації управління електронним бібліотечним фондом та обслуговування читачів у цифровому середовищі.

Предметом дослідження є методи та засоби розробки повностекового веб-застосунку бібліотеки з рольовою моделлю доступу, агрегацією зовнішніх книжкових каталогів і модулем персоналізованих рекомендацій на основі великої мовної моделі.

Метою дослідження є проєктування та реалізація веб-застосунку SmartLib, що забезпечує ведення каталогу книг, персоналізований пошук із залученням зовнішніх API, збереження читацьких нотаток, управління списком видань та генерацію AI-рекомендацій за жанром і переліком улюблених авторів.

Застосунок SmartLib реалізовано за клієнт-серверною архітектурою. Серверна частина побудована на платформі Node.js з фреймворком NestJS, що забезпечує модульну організацію коду та підтримку декораторів. Як реляційну СУБД обрано PostgreSQL; доступ до неї здійснюється через ORM Prisma, яка типізовано описує схему даних і автоматично генерує міграції. Клієнтська частина реалізована на бібліотеці React 18 з інструментом збирання Vite, Redux Toolkit для управління глобальним станом та RTK Query для декларативного виконання HTTP-запитів із вбудованим кешуванням [4].

Схема бази даних містить сутності Users (з підтримкою м'якого видалення через поле deletedAt), Profiles, Books, Orders, Notes (читацькі нотатки з прив'язкою до сторінки) та SavedBooks (збережені книги з ознакою джерела: PARSER або AI). Система розмежування доступу ґрунтується на трирівневій рольовій моделі (ADMIN, MANAGER, READER). Автентифікація реалізована за схемою JWT Bearer: при вході користувач отримує пару access/refresh-токенів, кожен захищений маршрут перевіряється JwtGuard, а RolesGuard зіставляє роль із необхідним рівнем доступу. Паролі зберігаються у вигляді bcrypt-хешу з коефіцієнтом складності 10. Додатково підтримується OAuth-авторизація через Google.

Модуль агрегації книжкових даних (books-parser) реалізує патерн Стратегія: інтерфейс BookProvider визначає контракт нормалізації, а два провайдери - GoogleBooksProvider та OpenLibraryProvider - звертаються до відповідних зовнішніх API і приводять відповіді до єдиного формату NormalizedBook, що дозволяє підключати нові каталоги без зміни бізнес-логіки.

Модуль AI-рекомендацій використовує OpenAI Chat Completions API (модель gpt-4o-mini) для генерації переліку книг за заданим жанром та улюбленими авторами. Промпт сконструйовано з примусовою відповіддю у форматі JSON-об'єкта (response_format: json_object). Структура відповіді містить поля similarGenres, books (масив рекомендацій з обґрунтуванням) та explanation. Для зниження витрат реалізовано серверне кешування

відповідей терміном 24 години за ключем із жанром, лімітом і відсортованим переліком авторів. Мережева стійкість забезпечена механізмом повторних спроб (до 2 ретраїв) з експоненційним відкладенням (базовий інтервал 800 мс); помилки вичерпання квоти та автентифікації не повторюються [5].

Висновки. У результаті виконаної роботи розроблено повностековий веб-застосунок SmartLib. Наукова новизна полягає у комплексній інтеграції трьох незалежних підсистем:

(1) агрегатора зовнішніх каталогів на основі патерну Стратегія, що нормалізує різнорідні джерела до єдиного формату;

(2) модуля AI-рекомендацій з 24-годинним серверним кешуванням та адаптивним retry-механізмом;

(3) трирівневої рольової системи управління доступом із підтримкою soft-delete та OAuth-авторизації.

Практичне значення роботи полягає у реалізації готового до розгортання застосунку, що автоматизує ключові процеси бібліотечного обслуговування. Застосування серверного кешування AI-відповідей дозволяє знизити кількість платних API-запитів, а нормалізований інтерфейс агрегатора забезпечує можливість розширення переліку підтримуваних каталогів без модифікації основної бізнес-логіки.

Список літератури

1. IFLA Trend Report 2023: Libraries in a Digital Age [Електронний ресурс]. — Режим доступу: https://www.ifla.org/trend-report (дата звернення: 20.05.2026).

2. Іванченко О. В. Сучасний стан цифровізації бібліотечних ресурсів України. Бібліотекознавство. Документознавство. Інформологія. 2024. № 2. С. 14–20.

3. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 3rd ed. Springer, 2022. 1008 p.

4. Abramov M. State management in modern React applications: Redux Toolkit and RTK Query. Journal of Web Engineering. 2023. Vol. 22, № 4. P. 531–558.

5. OpenAI Platform Documentation: Chat Completions API [Електронний ресурс]. — Режим доступу: https://platform.openai.com/docs (дата звернення: 20.05.2026).

Метлінський Д. А.
Західноукраїнський національний університет, магістрант
Науковий керівник д.т.н., проф. Пукас А.В.

РЕАЛІЗАЦІЯ ПІДСИСТЕМИ КВАЗІОПТИМІЗАЦІЇ СТРУКТУРИ КОМП'ЮТЕРНОЇ МЕРЕЖІ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ

Постановка проблеми. Наявність теоретичної багатокритеріальної математичної моделі є необхідною, проте недостатньою умовою для практичного проектування комп'ютерних мереж (КМ). Реальні топології містять десятки комутаційних вузлів та сотні зв'язків, що призводить до побудови матриць лінійних обмежень гігантської розмірності. Ручне обчислення таких систем за допомогою симплекс-методу чи М-методу є фізично неможливим і вимагає значних часових витрат. Відтак виникає гостра потреба у розробці спеціалізованого програмного забезпечення з високою обчислювальною швидкістю та гнучким монітор-діалоговим інтерфейсом, який би автоматизував розрахунок траєкторій поступок та наочно візуалізував результати для мережевого архітектора.

Мета роботи. Метою цієї частини роботи є проектування архітектури та безпосередня програмна реалізація комп'ютерної підсистеми багатокритеріальної оптимізації структур КМ у кросплатформному середовищі розробки Lazarus, а також проведення експериментального тестування створеного комплексу на прикладі модернізації регіональної IT-інфраструктури.

Основна частина. Для розробки комплексу обрано середовище візуального проектування Lazarus (мова Object Pascal, компілятор Free Pascal). Даний вибір зумовлений кросплатформністю середовища (можливістю запуску системи на операційних системах Windows та Linux без зміни вихідного коду), а також високою швидкодією згенерованого машинного коду під час виконання складних матричних ітерацій.

Архітектура створеного програмного забезпечення має чітку тримодульну структуру. Перший компонент — Препроцесор (MainUnit.pas) — реалізує інтерфейсну частину, приймає вхідні матриці суміжності, параметри надійності обладнання та автоматично будує початкову симплекс-таблицю. Другий компонент — Обчислювальний процесор (SimplexCore.pas) — є ядром системи, що покроково реалізує алгоритм послідовних поступок та М-метод великих штрафів для пошуку нових опорних планів. Третій компонент — Постпроцесор (PostProcessor.pas) — здійснює зчитування результуючих матриць, генерацію звітного текстового протоколу обчислень та графічну візуалізацію результатів (гістограм змін цільових функцій).

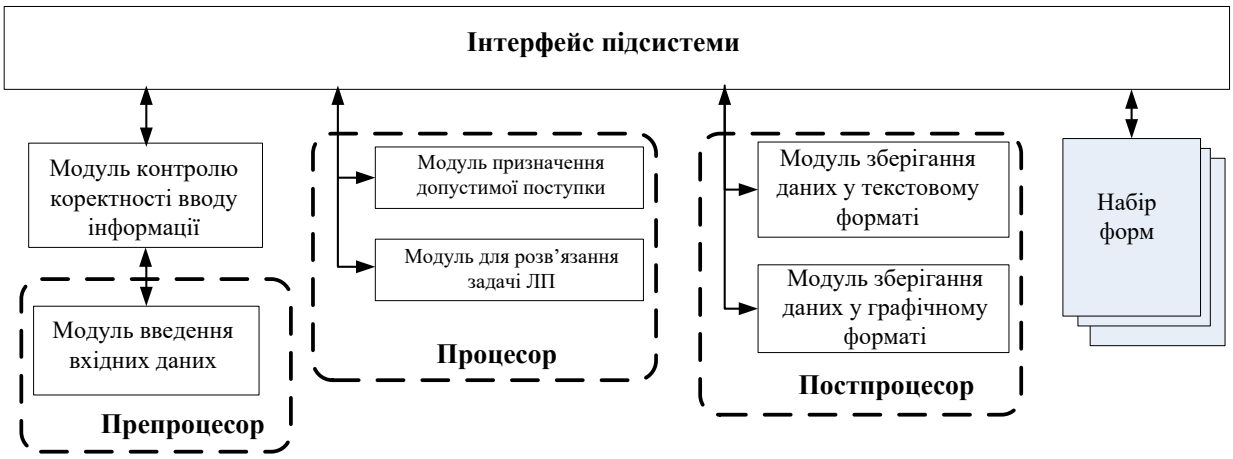
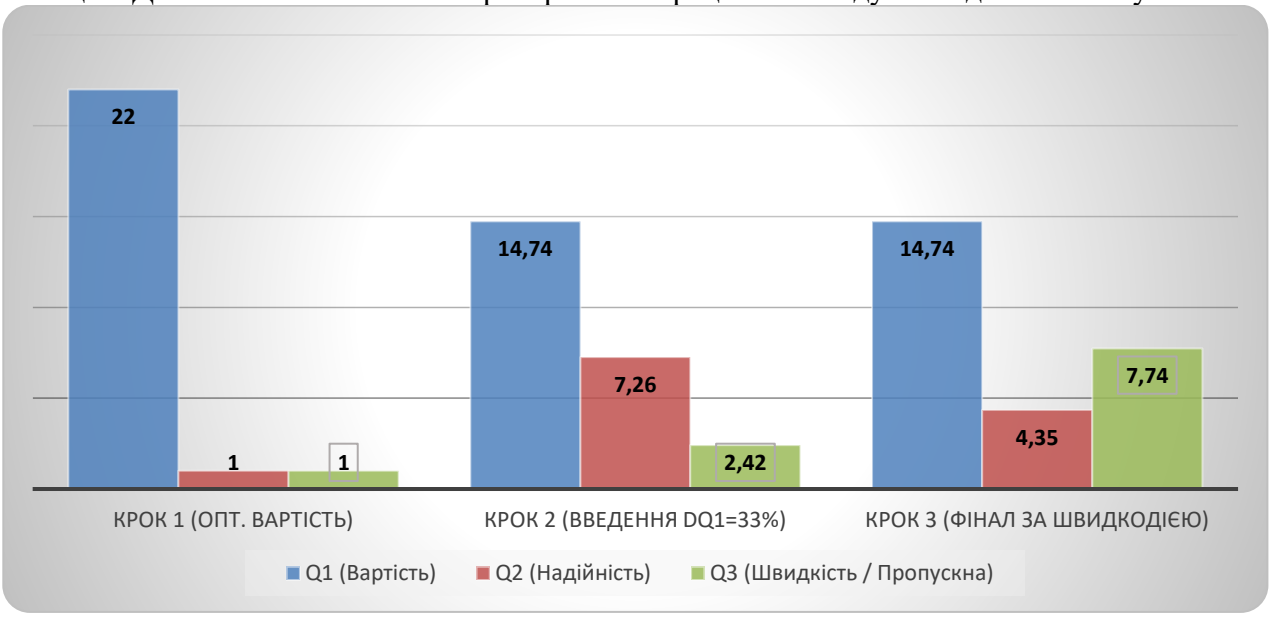


Рисунок 1 – Архітектурна схема підсистеми

Для верифікації системи проведено експериментальне дослідження на моделі регіональної комп'ютерної мережі, що складається з 15 вузлів. Оптимізація проводилася за

трьома критеріями: Q1 (вартість), Q2 (інтенсивність відмов) та Q3 (швидкодія). Траєкторія знаходження компромісного розв'язку за кроками алгоритму відображена в таблиці 1.

Таблиця 1 Динаміка зміни значень критеріїв за ітераціями методу послідовних поступок



На першому кроці було оптимізовано виключно фінансовий критерій, у результаті чого отримано мінімальну вартість Q1 = 22.0, проте показники надійності та швидкодії виявилися критично низькими. На другому кроці ОПР ввела компромісну поступку за вартістю у розмірі 33% (бюджет скориговано до 14.74). Обчислювальний процесор автоматично задіяв М-метод, перебудував систему обмежень та здійснив оптимізацію за критерієм надійності, покращивши показник Q2 до 7.26. На третьому кроці в межах утвореної області було максимізовано швидкодію. Фінальний компромісний план склав: Вартість = 14.74, Показник відмов = 4.35, Пропускна здатність = 7.74.

Аналіз експерименту підтвердив, що розроблене ПЗ дозволило обґрунтовано знизити середній час затримки пакетів у мережі на 23.5% та підвищити показник структурної надійності на 14.8%, повністю вклавшись у виділений бюджет модернізації інфраструктури. Обчислювальна швидкість виконання розрахунку ядра склала менше 1.8 секунди.

Висновки. Експериментальне тестування підтвердило повну працездатність та високу обчислювальну стійкість. Впровадження підсистеми дозволяє автоматизувати рутинні матричні розрахунки, мінімізувати вплив людського фактора при прийнятті інженерних рішень та знаходити точні компромісні топології комп'ютерних мереж. Розробка повністю готова до практичного впровадження як окремий інструментальний модуль у САПР-системи ІТ-підприємств.

Список використаних джерел

1. Dennis A., Wixom B. H., Roth R. M. Systems Analysis and Design / A. Dennis, B. H. Wixom, R. M. Roth. — Wiley, 2015. — 608 p.

2. Podeswa H. UML for the IT Business Analyst / H. Podeswa. — Course Technology, 2009. — 528 p.

3. George T. Heineman, Gary Pollice, Stanley Selkow. Algorithms in a Nutshell / G. T. Heineman, G. Pollice, S. Selkow. — O'Reilly Media, 2016. — 390 p.

4. T. L. Saaty, The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, McGraw-Hill, New York, 1980. — 287 p.

5. Michael T. Goodrich, Roberto Tamassia. Data Structures and Algorithms in Java/ M. T. Goodrich, R. Tamassia. — Wiley, 2014. — 736 p.

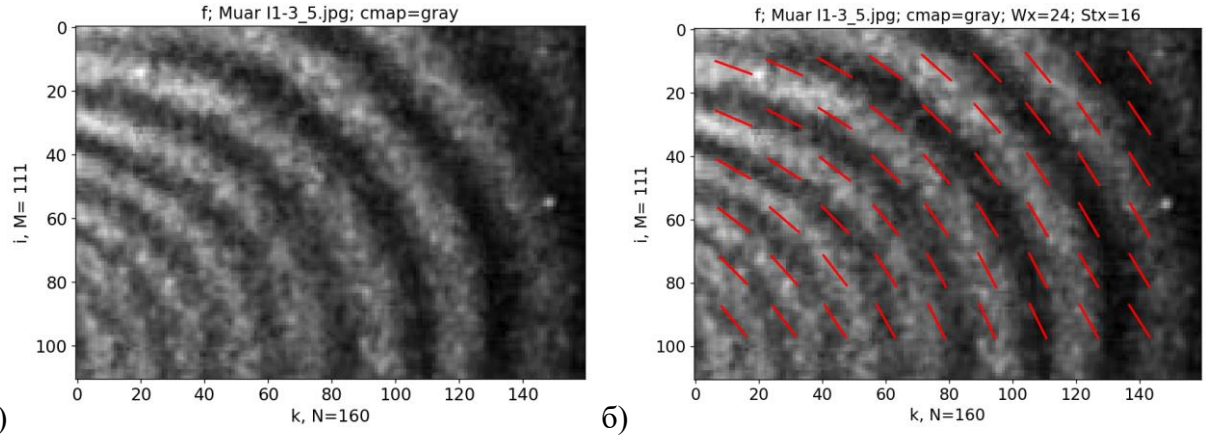
Яковлєв І.С.
аспірант 2 курсу Навчально-наукового інституту фізико-технічних та комп'ютерних наук Чернівецького національного університету імені Юрія Федьковича
Науковий керівник д.ф.-м.н., професор Фодчук І.М., кафедра інформаційних технологій та комп'ютерної фізики, Чернівецький національний університет імені Юрія Федьковича

ОРІЄНТОВАНА ФІЛЬТРАЦІЯ Х-ПРОМЕНЕВИХ ЗОБРАЖЕНЬ

Вступ. Завдяки високій проникаючі здатності Х-променів зображення, отримані з їх використанням, містять цінну інформацію про досліджувані зразки. Зокрема, Х-променеві муарові зображення дають змогу з надзвичайно високою точністю визначати деформації кристалів. У процесі цифрової обробки таких зображень проводиться аналіз періодів їх смуг у різних напрямах, що дозволяє визначити деформаційні поля для досліджуваних кристалів [1]. Проте, для експериментальних Х-променевих зображень характерні значні рівні шумів, що знижує точність їх подальшого аналізу. З метою зменшення рівня шумів виконується неорієнтована та орієнтована фільтрація зображень [2]. Орієнтована фільтрація особливо ефективна для зображень зі смугами, що є характерним для муарових зображень, оскільки у випадку фільтрації вздовж напряму смуг у значній мірі зберігається їх форма. Тому тематика даної роботи, яка пов'язана з орієнтованою фільтрацією Х-променевих зображень, є актуальною.

Постановка задачі. Метою даної роботи є розробка методики та програмного забезпечення на мові Python для орієнтованої фільтрації Х-променевих зображень з використанням орієнтованого фільтра Гауса. Методика обробки зображень повинна передбачати визначення напряму смуг у локальних прямокутних ділянках зображень, інтерполяцію значень кута орієнтації фільтра для всіх пікселів зображення та орієнтовану фільтрацію зображення.

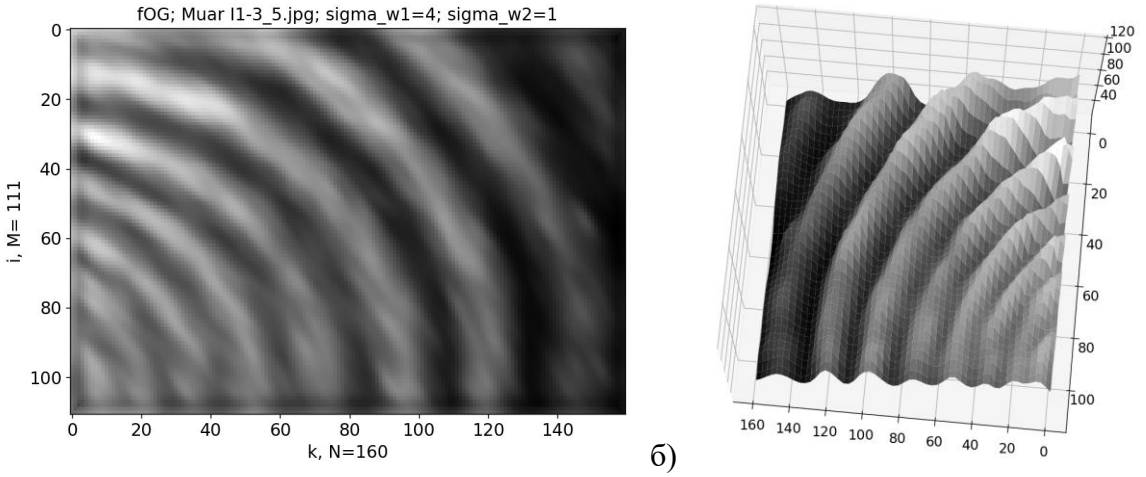
Основний матеріал. Ядро *wOG* двовимірного орієнтованого фільтра Гауса (розміром $Qu \times Qv$ елементів) описується кутом α, який визначає орієнтацію його головної осі, та середніми квадратичними відхиленнями (СКВ) σ_{w1} та σ_{w2} ядра вздовж та перпендикулярно до головної осі фільтра [3]. Програмна реалізація орієнтованої фільтрації зображень виконана на мові Python. Значення яскравостей початкових зображень *f* записувалися у масиви розміром $M \times N$ елементів. Локальна обробка зображень полягала в їх поділі на квадратні вікна розміром $Wx \times Wx$ пікселів, центри яких розміщувалися у вузлах прямокутної сітки у кроком *Stx* за висотою і шириною. Розмір вікон вибирався співрозмірним із шириною смуги. Для кожного вікна визначався середній напрям градієнту яскравості початкового зображення *f*, а кут α ядра фільтра обчислювався з урахуванням його перпендикулярності до напряму градієнту. Для орієнтованої фільтрації потрібні значення кута α для кожного пікселя зображення *f*, тому виконується бікубічна інтерполяція значень кута α.



а) б)

Рисунок 1 – Фрагмент експериментального Х-променевого муарового зображення *f* (а) та напрями орієнтації смуг (кут α) у межах вікон розміром $Wx \times Wx$ пікселів на зображенні *f* (б)

У процесі орієнтованої фільтрації зображення для кожного пікселя на основі кута α (який описує орієнтацію смуг) створюється ядро фільтра *wOG*. Шляхом згортки початкового зображення *f* з ядрами фільтрів *wOG* обчислюється фільтроване зображення *fOG*. Проведена орієнтована фільтрація експериментальних Х-променевих муарових зображень забезпечила значне зменшення рівня шумів при збереженні форми смуг (рис. 2).



а) б)

Рисунок 2 – Зображення після орієнтованої фільтрації: а) 2D візуалізація (як зображення); б) 3D візуалізація (як поверхня); СКВ орієнтованого ядра Гауса σw1 = 4, σw2 = 1

Після орієнтованої фільтрації смуги на зображенні чіткіше розділяються, що особливо важливо для смуг з малими просторовими періодами. Аналіз зображень смуг та їх профілів після орієнтованої фільтрації дає змогу з високою точністю визначити значення періодів смуг, оскільки на фільтрованих зображеннях смуги мають чіткі максимуми. Однією з проблем орієнтованої фільтрації є вибір розміру *Wx* квадратного вікна, у межах якого визначається напрям фільтрації. У даній роботі дана проблема вирішена шляхом перебору значень *Wx* у допустимому діапазоні (наприклад, від 8 до 64 пікселів) із заданим кроком. Для подальшої обробки використано напрями ядра фільтра для такого значення *Wx*, при якому напрями у сусідніх вікнах є найбільш подібними (оскільки смуги на Х-променевих муарових зображеннях є неперервними, тому й напрям фільтру повинен змінюватися плавно і відповідно до напряму смуги). Значна рівниця між напрямом фільтрів у сусідніх вікна, як правило, свідчить про невідповідність розміру вікна і періоду смуги. Оскільки на одному зображенні можуть бути присутніми смуги з різними періодами, тому наступним кроком запропонованої методики орієнтованої фільтрації є одночасне використання кількох розмірів вікон для локальних ділянок з різним періодом смуг.

Висновки. Розроблено методику та програму на мові Python для орієнтованої фільтрації Х-променевих зображень з використанням орієнтованого фільтра Гауса. Згідно з методикою напрям смуг у локальних вікнах зображень визначався на основі зміни градієнту яскравості зображення. У результаті орієнтованої фільтрації експериментальних Х-променевих муарових зображень значно зменшено їх рівні шумів при збереженні форми смуг. Аналіз періоду і напряму таких смуг дає змогу з високою точністю визначати деформації досліджуваних кристалів.

Список літератури

1. I.M. Fodchuk, S.V. Balovsyak, S.M. Novikov, I.V. Yanchuk, and V.F. Romankevych, "Reconstruction of spatial distribution of strains in crystals using the energy spectrum of X-ray Moire patterns, " *Ukrainian Journal of Physical Optics*, 2020. Vol. 21 (3), pp. 141-151.

2. L. Gengyi, "The Novel Bilateral Quadratic Interpolation Image Super-resolution Algorithm, " *International Journal of Image, Graphics and Signal Processing (IJIGSP*), 2021, Vol. 13 (3), pp. 55-61.

3. I. Fodchuk, S. Balovsyak, I. Iakovliev, T. Lytvynchuk, and I. Yanchuk, "Oriented filtering of X-ray Moire images," *Proceedings of SPIE: Seventeenth International Conference on Correlation Optics*, 2025, Vol. 13813, pp. 138133F-1 – 138133F-7.

Ілащук М.М.
аспірант 3 курсу Навчально-наукового інституту фізико-технічних та комп'ютерних наук Чернівецького національного університету імені Юрія Федьковича
Науковий керівник д.ф.-м.н., професор Мельничук С.В., кафедра комп'ютерних систем та мереж, Чернівецький національний університет імені Юрія Федьковича

РОЗПІЗНАВАННЯ ЕМОЦІЙ НА ЗОБРАЖЕННЯХ ОБЛИЧ У ВІДЕОПОТОЦІ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Вступ. Завдання розпізнавання емоцій на зображеннях облич є актуальним, оскільки результати такого розпізнавання широко застосовуються у різних сферах науки, техніки, освіти, медицини та бізнесу [1, 2]. Врахування емоцій користувача дає змогу підвищити ефективність людино-машинної взаємодії з використанням адаптивних інтерфейсів. В освіті емоційний стан студентів враховується для підвищення їх мотивації під час занять та індивідуалізації навчання. Проте, при розпізнаванні емоцій на реальних зображеннях облич виникають ускладнення, які пов'язані з потребою у детектуванні ділянки обличчя на зображенні та похибками розпізнавання емоцій. Крім цього, розпізнавання емоцій на зображеннях у відеопотоці потребує високої швидкодії, достатньої для обробки зображень у режимі реального часу. Тому у даній роботі розроблено систему розпізнавання емоцій на обличчях, яка задовольняє актуальні вимоги до таких систем розпізнавання.

Постановка задачі. Метою даної роботи є розробка системи розпізнавання емоцій на обличчях у відеопотоці в режимі реального часу. Початкові зображення облич отримувати як окремі графічні зображення або як кадри відеопотоку. Програмну реалізацію системи виконати на мові Python. Детектування прямокутних ділянок облич на зображеннях виконати нейронною мережею з архітектурою YOLO (You Only Look Once). Емоції на отриманих ділянках облич розпізнавати ансамблем згорткових нейронних мереж. З метою забезпечення функціонування системи реалізувати окремі потоки для зчитування зображень з відеокамери, їх візуалізації, детектування ділянок облич та розпізнавання емоцій на обличчях.

Основний матеріал. У розробленій програмі на мові Python виконується розпізнавання 7 базових емоцій: 1) Злість (Angry); 2) Огида (Disgust); 3) Страх (Fear); 4) Радість (Happy), 5) Нейтральність (Neutral); 6) Сум (Sad); 7) Здивування (Surprise) [3]. Безпосередньо розпізнавання емоцій виконано за допомогою моделі згорткової нейронної мережі (ЗНМ) [4], навченої на наборі даних FER-2013. Використано архітектуру ЗНМ, яка містить 6 згорткових шарів, а 2 вихідних шари є повнозв'язними. ЗНМ призначена для розпізнавання 7 емоцій, тому кожен з 7 виходів нейромережі означає ймовірність появи певної емоції на зображенні обличчя. На входи ЗНМ подаються зображення облич у відтінках сірого *Img*, масштабовані до розміру $M \times N$ пікселів методом бікубічної інтерполяції (наприклад, до розміру $M \times N = 48 \times 48$ пікселів у випадку використання набору даних FER-2013). Для підвищення точності розпізнавання емоцій облич, відповідно до запропонованої методики, застосовано не одну ЗНМ, а ансамбль з трьох ЗНМ [5]. Будова всіх ЗНМ №1-3 ансамблю однакова, а різниця полягає у вхідних сигналах.

Вхідним сигналом ЗНМ №1 є початкове зображення *Img* у відтінках сірого. На виходах ЗНМ №1 формується одновимірний масив $Y = (Y(j))$, де $j = 0,..., QY-1$, $QY = 7$ – кількість розпізнаних емоцій. Тобто значення елементів масиву $Y(j)$ означають ймовірність появи на вході ЗНМ обличчя, на якому розпізнається емоція з номером j. Таким чином розпізнається весь набір емоцій від злості ($j = 1$) до здивування ($j = 7$).

Вхідними сигналами ЗНМ №2 є зображення контурів *Cont*, які обчислюються на основі початкових зображень *Img* методом Собела. Як виходах ЗНМ №2 зчитується одновимірний масив *Ycont* розміром *QY* елементів, значення яких означають ймовірність розпізнавання певної емоції.

Вхідними сигналами ЗНМ №3 є зображення *Inv*, які обчислюються шляхом інверсії початкових зображень *Img*. На виходах ЗНМ №3 формується одновимірний масив *Ycont* розміром *QY* елементів, значення яких також означають ймовірність розпізнавання емоцій.

Вихідні масиви *Y*, *Ycont* та *Yinv* всіх ЗНМ ансамблю усереднюються, у результаті чого обчислюється одновимірний масив *YS* розміром *QY* елементів, значення яких означають ймовірність розпізнавання емоції на обличчі для всього ансамблю.

У програмі реалізовано 4 потоки (threads) для зчитування зображень з відеокамери, їх візуалізації, детектування ділянок облич та розпізнавання емоцій на обличчях. Потоки зчитування зображень та візуалізації виконуються у режимі реального часу (обробка 30 кадрів за секунду). В потоці детектування координати рамки обличчя визначаються попередньо навченою моделлю «yolov8n-face.pt» ЗНМ YOLO. Розпізнавання емоцій на обличчі виконується розробленим ансамблем ЗНМ. У потоці візуалізації на зображення накладаються актуальні рамки обличчя (отримані з потоку детектування) та найбільш ймовірні значення емоцій (отримані з потоку розпізнавання) (рис .1).

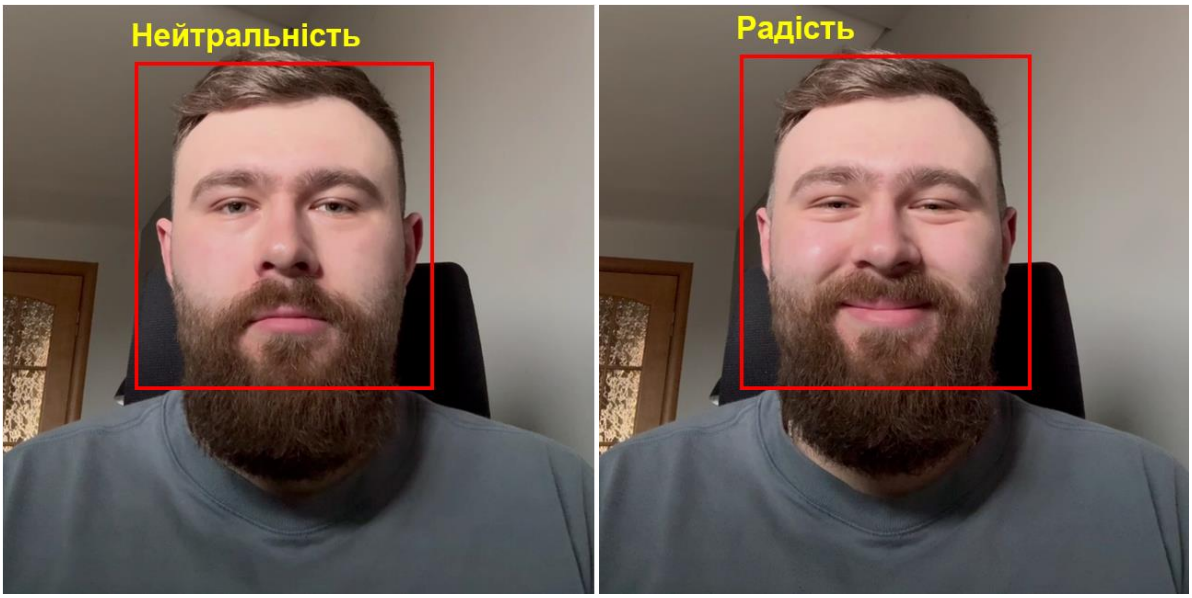


Рисунок 1 – Результати детектування облич на зображеннях та розпізнавання їх емоцій

Висновки. Розроблено програму на мові Python, призначену для розпізнавання емоції на обличчях у режимі реального часу. У програмі функціонує 4 потоки, що дозволяє узгодити високу швидкодію потоків зчитування зображень (30 кадрів за секунду) з нижчою швидкодією потоків детектування облич та розпізнавання емоцій (4 кадри за секунду). За рахунок використання ансамблю ЗНМ підвищено точність розпізнавання емоцій.

Список літератури

1. F.Z. Canal, T.R. Muller, J.C. Matias, G.G. Scotton, A.R. Junior, E. Pozzebon, and A.C. Sobieranski, "A survey on facial emotion recognition techniques: A state-of-the-art literature review, " *Information Sciences*, 2022, Vol. 582, pp. 593-617.

2. Ch. Liang, and J. Dong, "A Survey of Deep Learning-based Facial Expression Recognition Research, " *Frontiers in Computing and Intelligent Systems*, Vol. 5(2), pp. 56-60.

3. M. Chahed. Human Emotion Detection. URL: https://www.kaggle.com/code/mohamedchahed/human-emotion-detection.

4. O. Berezsky, P. Liashchynskyi, O. Pitsun, P. Liashchynskyi, and M. Berezkyy, "Comparison of Deep Neural Network Learning Algorithms for Biomedical Image Processing, " *CEUR Workshop Proceedings*, 2022, Vol. 3302, pp. 135-145.

5. O. Savenko, S. Balovsyak, O. Derevyanchuk, M. Ilashchuk, and S. Melnychuk, "Facial emotion recognition using an ensemble of convolutional neural networks, " *AIT&AIS'2025: International Scientific Workshop on Applied Information Technologies and Artificial Intelligence Systems*, December 18–19 2025, Chernivtsi, Ukraine, Vol. 4160, pp. 429-441.

Костюкевич Н.Ю.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

НЕЙРОМЕРЕЖЕВА СИСТЕМА ІНТЕРАКТИВНОГО ГОЛОСОВОГО ДІАЛОГУ З МУЗЕЙНИМИ ЕКСПОНАТАМИ НА ОСНОВІ АРХІТЕКТУРИ RAG

Вступ. Сучасний розвиток інформаційних технологій і систем штучного інтелекту суттєво змінює підходи до організації музейного простору. Водночас традиційні формати взаємодії з відвідувачами дедалі частіше демонструють свої обмеження. Класичні аудіогіди, сенсорні панелі та інформаційні таблички працюють у режимі односторонньої передачі інформації, де користувач фактично залишається пасивним споживачем заздалегідь підготовленого матеріалу. За такого підходу відсутня можливість поставити уточнювальне запитання й отримати відповідь у режимі реального часу [1, 2].

Поява великих мовних моделей (Large Language Models, LLM) та сучасних методів обробки природної мови (NLP) створила передумови для реалізації повноцінного природномовного діалогу безпосередньо в музейному середовищі. Однак пряме застосування LLM без опори на перевірені джерела інформації призводить до виникнення так званих «галюцинацій» — генераування переконливих, але фактично помилкових відповідей. Для культурно-освітньої сфери, у якій система сприймається як офіційне джерело даних, така поведінка є критично небажаною. Архітектура Retrieval-Augmented Generation (RAG) дає змогу усунути цю проблему завдяки жорсткій прив'язці генерації відповіді до фрагментів верифікованої бази знань [5].

Постановка задачі. Аналіз наявних рішень у сфері інтерактивних музейних AI-систем показав, що більшість із них реалізовано на закритих або комерційних платформах без відкритої технічної документації. Проєкти Museum of Zoology (Cambridge) та Artistic Chatbot (Warsaw Academy of Fine Arts) підтверджують актуальність голосового персонажного діалогу, проте не надають відтворюваного рішення, побудованого на сучасному NLP-стеку. Крім того, системи, орієнтовані виключно на текстову взаємодію або централізовані хмарні обчислення, не відповідають умовам роботи музейних сенсорних стендів, де GPU-обладнання зазвичай недоступне. Об'єктом дослідження є процес автоматизованої обробки природномовних голосових запитів у межах інтерактивних музейних систем. Предмет дослідження охоплює методи, моделі та програмні засоби побудови голосового RAG-конвеєра на базі відкритих NLP-технологій. Метою роботи є проєктування та реалізація нейромережевої системи інтерактивного голосового діалогу з музейними експонатами, здатної забезпечувати розпізнавання й синтез українського мовлення на CPU-обладнанні, семантичний пошук у верифікованій базі знань і генерацію відповіді від імені обраного персонажа.

Основний матеріал. Узагальнену архітектуру програмної системи спроєктовано як сукупність кількох взаємопов'язаних модулів.

Першим функціональним компонентом виступає модуль клієнтського інтерфейсу. Його основне призначення полягає у захопленні голосового введення відвідувача через браузерний MediaRecorder API та передачі бінарного аудіопотоку у форматі WebM на серверну частину за допомогою асинхронного HTTP-запиту. Клієнтський застосунок реалізовано як SPA на базі React 18 із використанням Vite для збирання та Tailwind CSS для стилізації інтерфейсу. AI-обробка на стороні клієнта не виконується. Фактично клієнтська частина виконує роль тонкого інтерфейсного шару, який також забезпечує відтворення MP3-файлу відповіді через стандартний браузерний Audio API.

Другим компонентом системи є модуль приймання та оркестрації запитів. Його реалізовано на основі асинхронного вебфреймворку FastAPI, що працює на ASGI-сервері Uvicorn і забезпечує неблокувальне виконання всіх I/O-операцій pipeline. Модуль приймає multipart-запит із аудіофайлом та ідентифікатором персонажа, маршрутизує його до відповідних сервісів і координує послідовність подальших етапів обробки. Центральним

менеджером конвеєра виступає фреймворк LangChain, який забезпечує уніфіковані механізми обміну даними між AI-компонентами системи.

Третім і ключовим компонентом є модуль розпізнавання мовлення (ASR). Його побудовано на бібліотеці faster-whisper - оптимізованій реалізації моделі Whisper від OpenAI на рушії CTranslate2. Модуль виконує транскрипцію аудіо виключно на CPU та не потребує GPU-прискорення. Використання int8-квантизації ваг моделі small (244 млн параметрів) дало змогу зменшити обсяг оперативної пам'яті та пришвидшити інференс без статистично значущого зниження точності розпізнавання українського мовлення. Перед етапом транскрипції WebM-аудіо обов'язково конвертується у формат WAV із частотою дискретизації 16 кГц засобами FFmpeg.

Четвертим компонентом виступає модуль семантичного пошуку (RAG). Розпізнаний текст запиту перетворюється на числовий вектор за допомогою embedding-моделі Sentence Transformers. Далі векторна база ChromaDB здійснює пошук найближчих фрагментів бази знань за метрикою косинусної подібності в іменованій колекції обраного персонажа. Ключовим елементом цього модуля є механізм перевірки релевантності: отриманий коефіцієнт подібності порівнюється з попередньо встановленим пороговим значенням. Якщо жоден із фрагментів не перевищує визначений поріг, система формує ввічливу відмову без звернення до LLM, що дає змогу превентивно блокувати появу галюцинацій. Саме цей підхід принципово відрізняє систему від прямого LLM-промптингу.

Алгоритм функціонування голосового RAG-конвеєру наведено на рисунку 1.

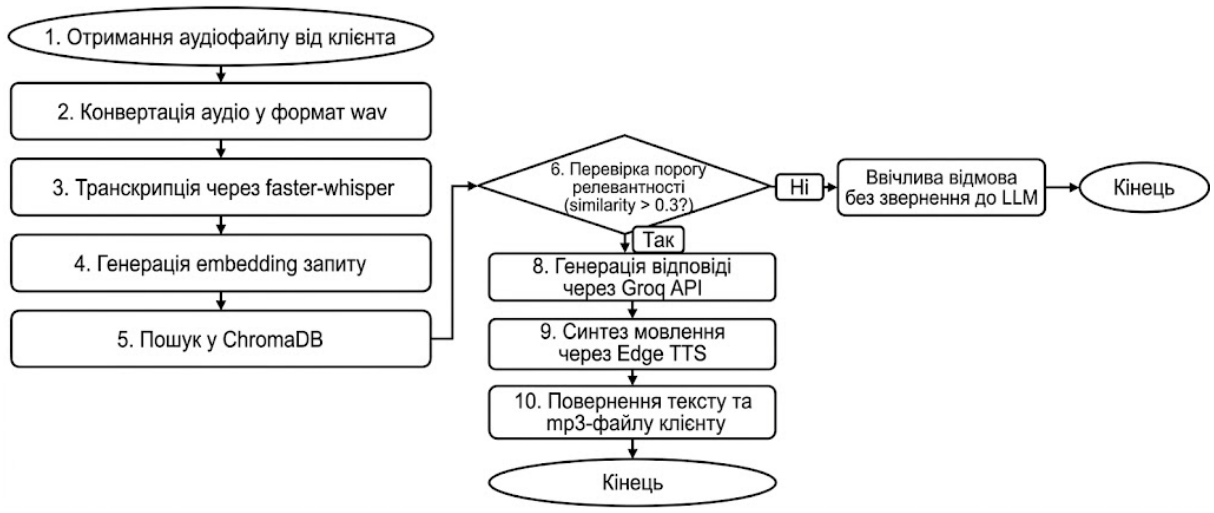


Рисунок 1 - Алгоритм функціонування голосового RAG-конвеєру

П'ятим компонентом є модуль генерації відповіді. Якщо семантичний пошук завершується успішно, знайдені фрагменти включаються до розширеного промпту разом із системним описом персонажа та запитанням користувача. Після цього промпт передається до генеративної моделі Llama 3.3 (70B) через хмарний inference API сервісу Groq, апаратна архітектура LPU якого забезпечує затримку генерації в межах 0.3–0.8 секунди. Використання параметрів temperature 0.7 та обмеження у 512 токенів дозволяє зберігати природність і варіативність відповіді без втрати стилістики персонажа [9].

Шостим компонентом є модуль синтезу мовлення (TTS). Згенерована текстова відповідь передається до бібліотеки Edge TTS, яка надає доступ до нейромережевих голосових рушіїв Microsoft Azure без необхідності оформлення платної підписки. Модуль асинхронно створює MP3-файл із природним українським мовленням голосом uk-UA-PolinaNeural та зберігає його у директорії статичних ресурсів із унікальним UUID-ідентифікатором. У відповідь клієнт отримує JSON-структуру з текстом відповіді та URL-адресою аудіофайлу для автоматичного відтворення. Повний час виконання pipeline на CPU-обладнанні становить у середньому 3–5 секунд.

Увесь програмний комплекс контейнеризовано за допомогою Docker та Docker Compose із виокремленням двох сервісів – фронтенду та бекенду. Векторна база ChromaDB зберігається між перезапусками через іменований том. Додавання нового персонажа не потребує внесення змін до програмного коду. Достатньо підготувати текстові документи та одноразово виконати скрипт індексування.

Висновки. У межах роботи спроєктовано модульну архітектуру системи, яка охоплює клієнтський інтерфейс захоплення голосу, серверну оркестрацію, модулі розпізнавання мовлення, семантичного пошуку у базі знань, генерації відповіді LLM та синтезу мовлення. Запропоновано реалізацію ASR-модуля на базі faster-whisper із використанням int8-квантизації, що забезпечує ефективну транскрипцію українського мовлення виключно на CPU-обладнанні.

Встановлено, що механізм порогового відсікання у модулі RAG є ключовим засобом запобігання галюцинаціям LLM. Під час тестування на запитаннях, відсутніх у базі знань, система жодного разу не сформувала сфабрикованої інформації. Загальний час відгуку в межах 3–5 с відповідає вимогам інтерактивних музейних стендів.

Список літератури

1. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W., Rocktäschel T., Riedel S., Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020. URL: https://arxiv.org/abs/2005.11401
2. Gao Y., Xiong Y., Gao X. та ін. Retrieval-Augmented Generation for Large Language Models: A Survey. 2024. URL: https://arxiv.org/abs/2312.10997
3. Radford A., Kim J. W., Xu T., Brockman G., McLeavey C., Sutskever I. Robust Speech Recognition via Large-Scale Weak Supervision. 2022. URL: https://arxiv.org/abs/2212.04356
4. Guillem Pages. faster-whisper: Fast Automatic Speech Recognition via CTranslate2. 2025. URL: https://github.com/SYSTRAN/faster-whisper
5. Rany2. edge-tts: Use Microsoft Edge's online text-to-speech service from Python without an official API. 2025. URL: https://github.com/rany2/edge-tts
6. Chase H. LangChain: фреймворк для розробки застосунків на основі LLM та технології Retrieval-Augmented Generation. 2025. URL: https://python.langchain.com
7. Groq Inc. Groq API Documentation: Llama 3.3 70B Versatile – LPU Inference Engine. 2025. URL: https://console.groq.com/docs/models
8. Chroma. ChromaDB: відкрита AI-нативна векторна база даних. 2025. URL: https://www.trychroma.com
9. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention Is All You Need. Advances in Neural Information Processing Systems. 2017. URL: https://arxiv.org/abs/1706.03762

Батько Ю.М.
к.т.н., доцент, кафедра КІ ЗУНУ

СИСТЕМИ МОНІТОРИНГУ УМОВ ПРОЖИВАННЯ НА ОСНОВІ ФІЗИЧНИХ ТА ХІМІЧНИХ ПАРАМЕТРІВ ЗОВНІШНЬОГО СЕРЕДОВИЩА В СТРУКТУРІ «РОЗУМНОГО МІСТА»

Вступ. Концепція «розумного міста» передбачає інтеграцію інформаційно-комунікаційних технологій, сенсорних мереж та аналітичних платформ для підвищення якості життя мешканців, ефективності управління міськими ресурсами та екологічної безпеки. Серед численних підсистем «розумного міста» особливе місце посідає моніторинг умов проживання, що забезпечує безперервне спостереження за фізичними та хімічними параметрами середовища. Такі системи є базовим інструментом для оцінки комфортності життя, прогнозування ризиків для здоров'я населення та обґрунтування містобудівних рішень [1].

Постановка задачі. Об'єкт дослідження – підходи та технології проєктування та розвитку «розумних міст». Предмет дослідження – алгоритми фіксації та оцінки основних параметрів навколишнього середовища в межах населених пунктів. Головна мета даного дослідження полягає в аналізі фізичних та хімічних характеристик навколишнього середовища в межах населених пунктів з метою обчислення рівня комфортності проживання. Що в подальшому дозволить здійснювати автоматизований моніторинг та прогнозування шляхів розвитку окремих елементів населених пунктів.

Вступ. Повнофункціональна система моніторингу в структурі «розумного міста» складається з трьох рівнів, а саме сенсорного, комунікаційного та аналітичного. Сенсорний рівень включає розподілену мережу стаціонарних та мобільних датчиків. Комунікаційний рівень ґрунтується на технологіях LoRaWAN, NB-IoT або 5G. Аналітичний рівень являє собою хмарну платформу з елементами машинного навчання. Просторове розташування вузлів моніторингу визначається топологією міської забудови, наявністю транспортних магістралей, промислових зон, зелених насаджень та житлових кварталів. Рекомендована щільність розгортання становить не менше одного багатофункціонального вузла на 0.5 квадратного кілометра у центральних районах та один вузол на 2 квадратні кілометри у спальних зонах. Час дискретизації для швидкозмінних параметрів, таких як загазованість та шум, дорівнює одній-десяти секундам, для температурно-вологісних характеристик – одній хвилині, а для добового усереднення тиску – десяти хвилинам [1].

Кожен з шести фізичних індикаторів розглядається як самостійний критерій, що має власну шкалу комфортності, гранично допустимі концентрації або еквівалентні рівні. Акустичний комфорт, що визначається рівнем зашумленості, є критичним для психофізіологічного стану людини. Система моніторингу фіксує еквівалентний рівень звуку з частотним зважуванням, яке імітує чутливість людського вуха. Оптимальна зона комфорту для житлових приміщень вдень відповідає рівню нижче 40 децибелів. Допустима зона для спальних районів та парків знаходиться в межах від 40 до 55 децибелів. Зона дискомфорту, що спостерігається поблизу магістралей у нічний час, становить від 55 до 65 децибелів, що підвищує ризик порушень сну та артеріальної гіпертензії. Критичний рівень вище 65 децибелів фіксується у промислових зонах та на вокзалах. Інтегральний показник акустичної комфортності розраховується як частка часу протягом доби, коли рівень не перевищує 55 децибелів у денний період та 45 децибелів у нічний.

Температурний режим оцінюється в приземному шарі атмосфери на висоті двох метрів над поверхнею з урахуванням сезонної адаптації людини. Використовується індекс ефективної температури та відносного температурного комфорту за шкалою ASHRAE. Для помірного кліматичного поясу комфортний діапазон становить від плюс 20 до плюс 25 градусів Цельсія при відносній вологості від 40 до 60 відсотків. Умовно дискомфортним вважається діапазон нижче плюс 18 градусів через ризик переохолодження при тривалому перебуванні на вулиці або вище плюс 28 градусів до плюс 32 градусів через тепловий стрес.

Відносна вологість повітря оцінюється в парі з температурою, оскільки суб'єктивне відчуття комфорту визначається тепловологісним індексом. Комфортний діапазон вологості становить від 40 до 60 відсотків за будь-якої комфортної температури. Допустимий діапазон включає значення від 30 до 40 відсотків та від 60 до 70 відсотків. У структурі «розумного міста» дані про вологість інтегруються з прогнозами вентиляції та системами зволоження або осушення в громадських будівлях.

Атмосферний тиск є метеотропним фактором, що впливає на самопочуття метеозалежних осіб. Хоча людина адаптується до середніх значень, різкі добові коливання мають діагностичне значення. Нормальним фоном вважається тиск у межах 760 +-11 міліметрів ртутного стовпчика, для даної місцевості з урахуванням висоти над рівнем моря. Комфортною динамікою є зміна тиску менше 2 гектопаскалів за 3 години. Моніторинг тиску є просторово-однорідним, тому достатньо одного референсного датчика на 5-10 квадратних кілометрів, доповненого даними з висотних будівель для виявлення перепадів по вертикалі.

Під терміном «загазованість» у міському моніторингу розуміють комплекс показників, що включає оксид вуглецю, діоксид азоту, діоксид сірки та приземний озон. Кожен з цих газів має власні гранично допустимі концентрації, проте для інтегральної оцінки комфортності використовується безрозмірний індекс якості повітря. Системи моніторингу фіксують не лише середні значення, а й пікові концентрації, наприклад оксиду вуглецю до 50 міліграмів під час заторів, що дозволяє оцінювати дискомфорт від короткочасних викидів.

Аерозольне забруднення у вигляді зважених часток є одним з найінформативніших критеріїв комфортності, оскільки дрібні частки проникають в альвеоли легень та кровотік. Для фракції $PM_{2,5}$, тобто часток діаметром менше 2.5 мікрометра, комфортним рівнем згідно з рекомендаціями ВООЗ є концентрація нижче 10 мікрограмів на кубічний метр, допустимим – від 10 до 15 мікрограмів, дискомфортним – від 15 до 35 мікрограмів, а критичним – вище 35 мікрограмів. Для фракції PM_{10}, тобто часток діаметром менше 10 мікрометрів, комфортним рівнем вважається концентрація нижче 20 мікрограмів на кубічний метр, допустимим – від 20 до 30 мікрограмів, дискомфортним – від 30 до 50 мікрограмів, а критичним – вище 50 мікрограмів. Оцінка комфортності також враховує співвідношення $PM_{2,5}$ до PM_{10}. Якщо це співвідношення перевищує 0.7, воно свідчить про переважання антропогенного дрібного пилу від двигунів та промисловості, що знижує комфортність більшою мірою, ніж крупний природний пил [2].

Для інтеграції зазначених критеріїв у єдину систему оцінки використовують математичні моделі, що дозволяють формувати узагальнений показник комфортності [3]. Перша формула може бути представлена як середньозважене значення параметрів:

$$\left[C_1 = \frac{w_n \cdot N + w_t \cdot T + w_h \cdot H + w_p \cdot P + w_g \cdot G + w_d \cdot D}{w_n + w_t + w_h + w_p + w_g + w_d}\right],$$

де N – рівень шуму, T – температура, H – вологість, P – атмосферний тиск, G – загазованість, D – рівень пилу, w – вагові коефіцієнти, що визначають значущість кожного параметра.

Дана формула базується на принципі середньозваженого значення, що дозволяє інтегрувати різні параметри у єдиний показник. Кожен параметр має власний ваговий коефіцієнт w, який визначає його значущість у загальній оцінці. Наприклад, шум та загазованість можуть отримати більшу вагу, ніж атмосферний тиск, оскільки їхній вплив на здоров'я є більш критичним. Таким чином, формула дозволяє адаптувати систему оцінки до конкретних умов міста, враховуючи пріоритетність окремих факторів.

Також використовують формулу нелінійного впливу окремих факторів:

$$\left[C_2 = \alpha \cdot e^{-\beta N} + \gamma \cdot f(T,H) + \delta \cdot \frac{1}{1+G+D}\right].$$

Функція $f(T,H)$описує взаємозалежність температури та вологості, адже комфортність визначається не лише окремими значеннями, а й їхньою комбінацією. Наприклад, висока

температура при низькій вологості може бути прийнятною, тоді як висока температура при високій вологості створює значний дискомфорт. Останній доданок враховує загазованість та пил, де навіть невелике їхнє зростання може суттєво знизити якість життя.

Ще одна формула може бути побудована на основі нормалізації параметрів:

$$\left[C_3 = \frac{1}{6}\left(\frac{N_{opt}}{N} + \frac{T_{opt}}{|T - T_{opt}| + 1} + \frac{H_{opt}}{|H - H_{opt}| + 1} + \frac{P_{opt}}{|P - P_{opt}| + 1} + \frac{1}{1+G} + \frac{1}{1+D}\right)\right].$$

Формула використовує нормалізацію параметрів відносно їхніх оптимальних значень. Вона дозволяє оцінити, наскільки фактичні умови відхиляються від комфортних. Чим ближче значення параметра до оптимального, тим більший внесок він робить у загальний показник. Наприклад, якщо температура дорівнює оптимальній, відповідний доданок наближається до максимального значення. Якщо ж температура значно відхиляється, внесок зменшується. Такий підхід забезпечує більш точну оцінку, оскільки враховує не лише абсолютні значення, а й їхню відповідність нормам.

Наступна формула може враховувати інтегральний індекс екологічного комфорту:

$$\left[C_4 = \sqrt{\frac{(N_{norm}^2 + T_{norm}^2 + H_{norm}^2 + P_{norm}^2 + G_{norm}^2 + D_{norm}^2)}{6}}\right].$$

Наведена формула базується на концепції інтегрального індексу, де всі параметри нормалізуються у діапазоні від 0 до 1. Використання квадратів дозволяє підсилити вплив значних відхилень, адже навіть невелике перевищення норми може суттєво знизити загальний показник. Формула відображає середнє квадратичне значення параметрів, що забезпечує більш збалансовану оцінку. Вона особливо корисна для систем, де необхідно враховувати комплексний вплив усіх факторів одночасно.

Системи моніторингу умов проживання можуть бути інтегровані у структуру «розумного міста» для забезпечення постійного контролю за якістю життя. Вони можуть використовувати мережі сенсорів, розташованих у різних районах міста, для збору даних у реальному часі. Отримані дані можуть передаватися до центральної системи, яка здійснює їх аналіз та формує індекс комфортності. Це дозволяє органам місцевого самоврядування приймати обґрунтовані рішення щодо покращення інфраструктури, зменшення рівня шуму, зниження забруднення та оптимізації мікроклімату. Крім того, мешканці можуть отримувати інформацію про умови проживання через мобільні додатки, що сприяє підвищенню їхньої обізнаності та залученості.

Висновки. Системи моніторингу умов проживання у структурі «розумного міста» є важливим інструментом для забезпечення високої якості життя мешканців. Використання критеріїв, таких як рівень шуму, температура, вологість, тиск, загазованість та рівень пилу, дозволяє формувати комплексний показник комфортності. Запропоновані формули забезпечують інтеграцію різних параметрів та враховують їхній вплив на здоров’я та самопочуття. Практичне застосування таких систем сприяє підвищенню ефективності управління міською інфраструктурою та створенню сприятливих умов для життя.

Список літератури

1. Wei Liao, Dixin Wu Bo Pan, Congyue Qi and ets. Assessing the Integrated Impacts of Outdoor Thermal *Environment and Air Quality on Thermal Comfort in Residential Areas: A Case Study of Wuhan*. Buildings 2025, 15(23), 4309. *https://doi.org/10.3390/buildings15234309.*

2. World Health Organization, *Environmental Health Indicators for Urban Areas*, WHO Press, Geneva, 2024.

3. European Environment Agency, *Air Quality and Urban Comfort Indices*, EEA Report, Copenhagen, 2025.

Івашенюк С.О.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Загородня Д.І., кафедра ІОСУ ЗУНУ

РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ КЕРУВАННЯ ПРОЄКТНИМИ ЗАВДАННЯМИ

Вступ. Сучасні компанії та команди дедалі частіше застосовують цифрові рішення для організації та управління спільною роботою. Особливу важливість має питання ефективного керування завданнями у невеликих командах, успіх проєктів яких значною мірою залежить від оперативного обміну даними, контролю за дотриманням строків та розподілу функцій між учасниками. Застосування веб-додатків дозволяє автоматизувати ці механізми, забезпечуючи єдиний доступ до даних та можливість колективної роботи в реальному часі [1-4].

Постанова задачі. Основна мета даної роботи полягає у створенні онлайн-системи для управління проєктними завданнями у невеликих робочих колективах, що дозволить автоматизувати ключові етапи планування, розподілу та відстеження виконання робіт. Об'єктом дослідження є процес організації та управління завданнями в контексті малих робочих груп. Предметом дослідження виступають методології, моделі та програмні засоби, що використовуються для реалізації веб-застосунків, призначених для керування та моніторингу виконання завдань.

Основний матеріал. В рамках даного дослідження було проведено детальний аналіз існуючих систем управління проєктами та завданнями, зокрема таких популярних платформ, як Trello, Asana та Monday.com. Дослідження охоплювало вивчення їхніх функціональних можливостей, інтерфейсних рішень, механізмів організації командної взаємодії, планування та контролю виконання завдань. Особливу увагу було приділено інструментам управління життєвим циклом проєктів, засобам моніторингу прогресу, системам сповіщень, інтеграції з зовнішніми сервісами та можливостям візуалізації даних [5]. За результатами проведеного аналізу було визначено основні переваги та обмеження існуючих рішень, ідентифіковано найбільш затребувані функції для ефективної організації роботи користувачів, а також сформульовано функціональні та нефункціональні вимоги до розроблюваного програмного продукту. Отримані результати стали основою для проєктування архітектури системи та вибору оптимальних технологічних рішень для її реалізації.

Загальна структура програми містить клієнтську і серверну ланки та сховище даних. Клієнтська ланка (Frontend) забезпечує взаємодію користувача із програмою через веб-інтерфейс. Серверна ланка (Backend) виконує опрацювання запитів користувачів та реалізує бізнес-логіку. Сховище даних забезпечує збереження та опрацювання відомостей про користувачів, проєкти та доручення.

Головні функціональні блоки програми:

– блок керування користувачами. Забезпечує: реєстрацію користувачів, вхід у програму, керування особистими даними користувачів;

– блок керування проєктами. Забезпечує: створення проєктів, зміну відомостей про проєкт, додавання учасників до проєкту;

– блок керування дорученнями. Забезпечує: створення доручень, призначення виконавців, встановлення крайніх термінів, зміна стану виконання;

– блок комунікацій. Забезпечує: додавання коментарів до доручень, обмін файлами, сповіщення між учасниками.

Інформаційні зв’язки між частинами програми забезпечуються через передачу даних між клієнтською ланкою, сервером та сховищем даних.

Для проєктування програми застосовано об’єктно-орієнтований метод, який дозволяє представити програму як взаємодію об’єктів предметної сфери. Основними об’єктами програми є: user, project, task, comments та file. Кожен об’єкт має набір ознак та методів, що окреслюють його функціонування у програмі.

Створений веб-застосунок надає функціонал для ефективного управління проєктами та завданнями з використанням Kanban-підходу. Система дозволяє формувати робочі області,

створювати проєкти та задачі, змінювати їхні статуси шляхом переміщення між стовпцями Kanban-дошки, призначати відповідальних виконавців та відстежувати хід виконання робіт. На рисунку 1 представлений інтерфейс розробленої системи для керування завданнями.

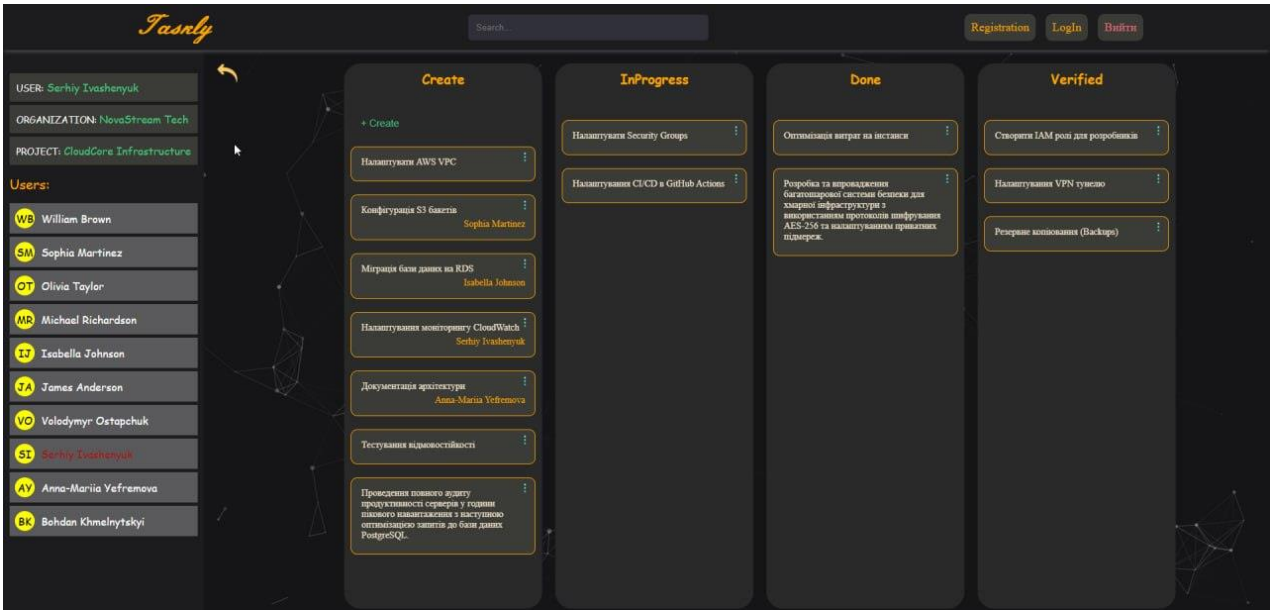


Рисунок 1 – Інтерфейс системи керування завданнями

Клієнтська частина програми реалізована за допомогою фреймворку Nuxt 3, що забезпечує високу швидкодію, зручну архітектуру компонентів та підтримку актуальних стандартів розробки веб-інтерфейсів[3]. Серверна частина побудована на мікросервісній архітектурі з використанням NestJS та GraphQL Gateway, що гарантує масштабованість системи та спрощує подальше розширення її функціональних можливостей.[1]

Для зберігання даних використовується система керування базами даних PostgreSQL[2]. Вона забезпечує надійне збереження інформації, підтримує складні взаємозв'язки між сутностями та демонструє високу продуктивність при роботі зі значними обсягами даних.

Розроблена система підтримує механізми автентифікації та авторизації користувачів, функціонал створення та редагування завдань, їх групування за статусами, а також сприяє ефективній взаємодії між учасниками робочих груп. Застосування Kanban-дошки дозволяє наочно відображати поточний стан виконання проєкту та сприяє підвищенню загальної ефективності командної роботи.

Висновки. У результаті проведеного дослідження було проаналізовано сучасні системи управління проєктами та завданнями, визначено їхні переваги, недоліки та основні функціональні можливості. На основі отриманих результатів сформовано вимоги до розробки веборієнтованої системи керування завданнями для невеликих робочих колективів. Спроєктовано архітектуру програмного забезпечення, що включає клієнтську та серверну частини, а також систему зберігання даних. Програма надає централізоване керування відомостями про користувачів, доручення, зауваження, часові межі виконання та стани робіт. Вона дає змогу організувати дієву взаємодію між членами команди, полегшує процес планування роботи та посилює нагляд за виконанням визначених завдань..

Використання технологій Nuxt 3, NestJS, GraphQL та PostgreSQL забезпечило високу продуктивність, масштабованість і надійність системи. Запровадження Kanban-підходу дозволило візуалізувати стан виконання завдань і підвищити ефективність командної взаємодії. Розроблений програмний продукт може бути використаний для автоматизації управління проєктною діяльністю малих робочих груп та слугувати основою для подальшого розширення функціональних можливостей системи.

Список літератури

1. NestJS. (2026). *NestJS Documentation*. Retrieved May 8, 2026, from https://docs.nestjs.com

2. PostgreSQL Global Development Group. (2026). *PostgreSQL Official Documentation*. Retrieved May 30, 2026, from https://www.postgresql.org/docs/

3. Nuxt.js Team. (2026). *Nuxt Documentation*. Retrieved May 8, 2026, from https://nuxt.com/docs

4. Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. Retrieved May 8, 2026, from https://www.pmi.org/pmbok-guide-standards/foundational/pmbok

5. Vue.js Team. (2026). *Vue.js Documentation*. Retrieved May 8, 2026, from https://vuejs.org/

Наконечний М.В.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Осолінський О.Р., кафедра ІОСУ ЗУНУ

ВИСОКОПРОДУКТИВНИЙ МОДУЛЬ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ ПОТОКОВОГО ВІДЕО

Вступ. Сучасні системи відеоспостереження, транспортного моніторингу, виробничого контролю та робототехніки формують значні обсяги відеоданих, які потрібно аналізувати безпосередньо під час надходження. Передавання повного потоку до віддаленого сервера не завжди є доцільним, тому що це збільшує затримку, навантажує мережу та ускладнює автономну роботу системи. Тому все більше звертають увагу на граничну відеоаналітика, в якій основні етапи обробки виконуються поблизу джерела даних [1]. Для таких задач доцільно використовувати компактні апаратні платформи з графічним прискоренням, зокрема Nvidia Jetson, а також засоби перенесення та оптимізації моделей, такі як ONNX і TensorRT [2, 3].

Актуальність роботи полягає в потребі створення компактного модуля комп'ютерного зору, здатного приймати відеопотік, виконувати підготовку кадрів, запускати модель виявлення об'єктів, формувати результати та подавати їх користувачу в зручному вигляді..

Постанова задачі. Метою роботи є розробка високопродуктивного модуля комп'ютерного зору для потокового відео, орієнтованого на використання в edge-середовищі та подальше розгортання на платформі Nvidia Jetson. Об'єктом дослідження є процес потокової відеоаналітики в граничних системах комп'ютерного зору. Предметом дослідження є методи, алгоритми та програмні засоби приймання, обробки, аналізу й представлення результатів відеопотоку.

Основний матеріал. В процесі дослідження розглянуто сервісні фреймворки для edge-відеоаналітики, рішення на базі Nvidia Jetson і засоби апаратно-орієнтованого прискорення інференсу. Аналіз показав, що практична ефективність системи залежить не лише від вибору моделі, але також від організації повного конвеєра обробки [4].

Розроблена структура модуля передбачає роботу з кількома типами джерел: IP-камерою, USB-камерою, RTSP-потоком або локальним відеофайлом.

Після підключення джерела кадри проходять декодування, масштабування, нормалізацію та перетворення у формат, придатний для подання на модель виявлення об'єктів. Центральним елементом є backend виконання моделі. В проекті передбачено модульний підхід: поточний стенд використовує mock_backend.py для відпрацювання повного конвеєра, а окремий

tensorrt_backend.py закладає можливість подальшого виконання оптимізованої моделі через TensorRT.

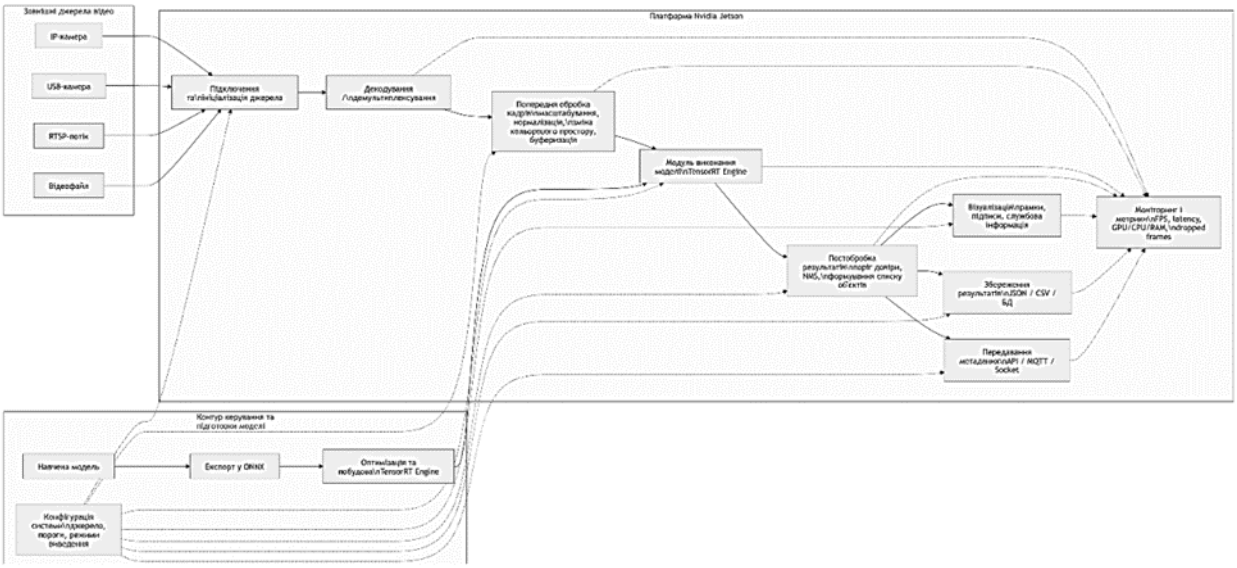


Рисунок 1 – Загальна структура модуля потокової відеоаналітики на платформі Nvidia Jetson

Програмна реалізація виконана в Linux-середовищі мовою Python. Для побудови веб-інтерфейсу використано Flask, для роботи з відеоданими - OpenCV, для числових операцій - NumPy, а для винесення налаштувань у зовнішній файл - PyYAML.

Функціональна логіка реалізована у вигляді окремих модулів. Файл main.py відповідає за маршрути веб-застосунку, завантаження відео та запуск обробки. Модуль video_source.py відкриває відеопотік і визначає його параметри. У pipeline.py зосереджено основну послідовність обробки кадрів: читання, виконання backend-модуля, нанесення рамок і формування підсумкової статистики. Модуль visualize.py відповідає за побудову анотованих кадрів, а storage.py зберігає вихідне відео, структуровані результати у JSON і метрики продуктивності у CSV.

Для взаємодії з користувачем створено тестову веб-сторінку, на якій можна завантажити відеофайл, переглянути службові параметри потоку, виконати попередній аналіз першого кадру та запустити повну обробку. Після завантаження відео система відображає розмір кадру, частоту кадрів, загальну кількість кадрів, кількість попередньо виявлених об’єктів і анотований кадр з рамками та підписами. Приклад інтерфейсу після завантаження тестового відео наведено на рисунку 2.

Під час тестування перевірено створення всіх вихідних файлів: анотованого відео у форматі MP4, файла detections.json зі структурованими результатами та файла metrics.csv з числовими показниками. В межах одного запуску було оброблено 150 кадрів, сформовано 300 виявлень, а повний час обробки становив 1,06 с.

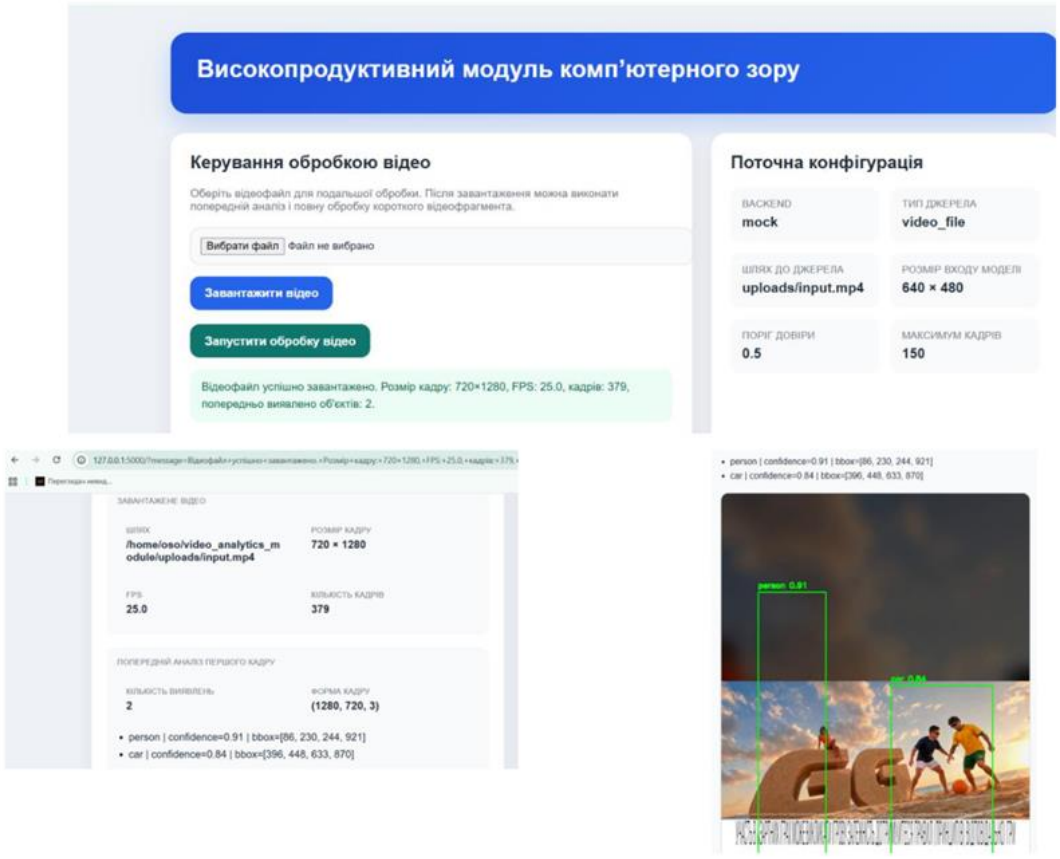


Рисунок 2 – Web-інтерфейс після завантаження відеофайлу та попереднього аналізу кадру

Висновки. В результаті було розроблено програмний стенд високопродуктивного модуля комп'ютерного зору для потокового відео. Створений веб-інтерфейс забезпечує завантаження відеофайлу, запуск аналізу, відображення попередніх результатів і перегляд активної конфігурації. Перевірка показала, що модуль коректно обробляє тестовий відеофрагмент, формує анотоване відео, JSON-файл виявлень і CSV-файл метрик. Розроблене рішення може бути використане в задачах відеоспостереження, транспортного моніторингу, промислового контролю та навчальних досліджень з граничних обчислень.

Список літератури

1. Hu M., Luo Z., Pasdar A., Lee Y. C., Zhou Y., Wu D. Edge-Based Video Analytics: A Survey. arXiv preprint. 2023. arXiv:2303.14329.
2. NVIDIA Jetson Linux Developer Guide. Quick Start. URL: https://docs.nvidia.com/jetson/archives/r35.6.1/DeveloperGuide/IN/QuickStart.html (дата звернення: 05.04.2026).
3. NVIDIA TensorRT. Quick Start Guide. URL: https://docs.nvidia.com/deeplearning/tensorrt/latest/getting-started/quick-start-guide.html (дата звернення: 21.02.2026).
4. Liu P., Qi B., Banerjee S. EdgeEye: An Edge Service Framework for Real-Time Intelligent Video Analytics. EdgeSys. 2018. P. 1-6. DOI: 10.1145/3213344.3213345.
5. Basulto-Lantsova A., Padilla-Medina J. A., Perez-Pinal F. J., Barranco-Gutierrez A. I. Performance comparative of OpenCV Template Matching method on Jetson TX2 and Jetson Nano developer kits. CCWC. 2020. P. 0812-0816. DOI:10.1109/CCWC47524.2020.9031166.

Калашнюк В.Б.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник викладач Шпак М.Ю., кафедра КІ ЗУНУ

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЛОКАЛЬНИХ БАЗ ДАНИХ У ПОРІВНЯННІ З ХМАРНИМИ API ДЛЯ СИСТЕМ АВАРІЙНИХ СПОВІЩЕНЬ РОЗУМНОГО БУДИНКУ

Вступ. Розвиток концепції Інтернету речей (IoT) та систем житлової автоматизації висуває нові вимоги до архітектури збору, обробки та візуалізації даних. Традиційний підхід, що базується на використанні зовнішніх хмарних інфраструктур та хмарних API, забезпечує швидке розгортання, проте створює критичні ризики для систем безпеки через залежність від стабільності інтернет-каналу. В задачах детекції аварійних ситуацій ключовим фактором є швидкість реакції та автономність системи. Завдяки перенесенню конвеєра обробки даних на локальні бази даних, досягається максимальна відмовостійкість програмних тригерів та мінімізація затримок при доставці сповіщень на вебпортал керування, що є критично важливим для збереження майна та життєдіяльності людини. [1]

Постановка задачі. Об’єкт дослідження — підходи до організації конвеєрів збору, обробки та синхронізації даних між сенсорними IoT-вузлами та веб-інтерфейсами керування. Предмет дослідження — порівняльна ефективність локальних вбудованих баз даних та зовнішніх хмарних API у системах аварійного сповіщення розумного будинку. Головна мета даного дослідження полягає в аналізі та обґрунтуванні архітектурних рішений для автономного вебпорталу безпеки, що забезпечує мінімальну затримку обробки критичних параметрів, зокрема рівня газу, та гарантує стабільність роботи тригерів сповіщення у разі відсутності підключення до глобальної мережі Інтернет.

Основний матеріал. Інтернет речей (IoT) зараз проникає в наше повсякденне життя, надаючи важливі інструменти вимірювання та збору даних, щоб обґрунтувати кожне наше рішення. Мільйони датчиків і пристроїв безперервно виробляють дані та обмінюються важливими повідомленнями через складні мережі, що підтримують міжмашинний зв'язок, а також моніторять і контролюють критично важливі інфраструктури розумного світу. Як стратегія пом'якшення ескалації перевантаження ресурсів, периферійні обчислення стали новою парадигмою для вирішення потреб IoT та локалізованих обчислень. Порівняно з добре відомими хмарними обчисленнями, периферійні обчислення перенесуть обчислення або зберігання даних на «периферію» мережі, ближче до кінцевих користувачів. [2]

У контексті систем житлової автоматизації, зокрема підсистем аварійної безпеки та сповіщення, переваги периферійних обчислень набувають вирішального значення. Традиційні архітектурні рішення, що покладаються на централізовані хмарні сервіси та зовнішні прикладні програмні інтерфейси, демонструють ряд критичних недоліків при обробці динамічних подій, таких як витік горючих газів чи задимлення приміщень. Основними деструктивними факторами хмарної моделі в задачах екстреного моніторингу є непередбачувана затримка доставки пакетів даних через нестабільність інтернет-каналу, обмеження пропускної здатності безкоштовних хмарних шлюзів, а також ризик повної втрати контролю над об'єктом у разі збою на стороні провайдера. Це зумовлює необхідність дослідження та розробки автономних конвеєрів збору телеметрії, де обробка даних та логіка спрацювання тригерів небезпеки реалізуються безпосередньо всередині локальної мережі за допомогою інтеграції вбудованих баз даних.

Процес передачі даних доцільно розглянути стандартний конвеєр взаємодії IoT-пристроїв із хмарними платформами. У такій моделі сенсорний вузол здійснює періодичний запис телеметрії за допомогою HTTP-запитів типу POST до віддаленого сервера. Вебпортал користувача, у свою чергу, функціонує за принципом періодичного опитування (polling) хмарного сховища для оновлення графіків та індикаторів стану.

Головною системною проблемою такої конфігурації є штучно створена затримка обміну даними (data latency). Більшість відкритих хмарних API мають жорсткі ліміти пропускної здатності для некомерційного використання, що дозволяють надсилати запити з інтервалом не частіше ніж один раз на 15–20 секунд. У ситуації, коли рівень концентрації газу або задимлення

зростає лавиноподібно, затримка у 20 секунд між фіксацією небезпеки та її відображенням на вебпорталі може стати критичною. Крім того, будь-яка деградація інтернет-каналу або технічні збої на стороні хмарного провайдера повністю ізолюють вебпортал від даних, роблячи систему моніторингу непрацездатною.

Альтернативним рішенням, що усуває вищеописані недоліки, є повна локалізація контуру обміну інформацією в межах домашньої мережі без залучення зовнішніх інтернет-ресурсів. У такій конфігурації проміжним середовищем збереження даних виступає локальна вбудована база даних, розгорнута безпосередньо на периферійному шлюзі або керуючому сервері розумного будинку.

Для комплексного оцінювання та наочного співставлення розглянутих підходів було сформовано систему критеріїв ефективності. Узагальнені результати порівняльного аналізу хмарної (Cloud-based API) та локальної (On-Premises) архітектур наведено в таблиці 1.

Таблиця 1 - Порівняльний аналіз архітектурних рішень для систем аварійних сповіщень

Критерій порівняння	Хмарне API (Cloud-based)	Локальна БД (On-Premises)
Швидкість оновлення даних	Обмежена лімітами API (15–20 сек)	Обмежена лише швидкістю диска (мілісекунди)
Залежність від Інтернет-каналу	Критична (при обриві система не працює)	Відсутня (повна автономність у локальній мережі)
Складність розгортання	Низька (готові хмарні конструктори)	Середня (потребує налаштування локального сервера)
Конфіденційність приватності	Ризик витоку даних на сторонніх серверах	Максимальна (дані зберігаються локально)
Надійність тригерів аварії	Залежить від стабільності серверів хмари	Абсолютна в межах локального заліза

Детальний аналіз наведених у таблиці 1 даних дозволяє зробити висновок, що за більшістю критично важливих параметрів локальна On-Premises архітектура суттєво перевершує хмарні аналоги. Зокрема, у контексті швидкості оновлення даних локальний підхід демонструє чисельну перевагу, оскільки час виконання транзакцій у сучасних вбудованих базах даних вимірюється мілісекундами, тоді як безкоштовні хмарні шлюзи штучно обмежують частоту запитів до 15–20 секунд. Така різниця у часі відгуку є визначальною для мінімізації наслідків надзвичайних ситуацій. Аналіз надійності тригерів аварії та залежності від мережевих каналів чітко вказує на деструктивну роль глобального Інтернету для систем безпеки. У разі аварії на стороні провайдера хмарна модель повністю втрачає комунікацію, тоді як локальна база даних забезпечує стовідсоткову автономність контуру моніторингу всередині домашньої мережі. Єдиним компромісним показником локальної архітектури є дещо вища складність розгортання, оскільки вона вимагає від розробника самостійного конфігурування локального сервера (наприклад, на базі програмних скриптів та СКБД) замість використання готових хмарних конструкторів. Проте цей недолік повністю нівелюється досягненням максимального рівня конфіденційності приватності користувача, оскільки персональна телеметрія житла гарантовано не передається на сервери третіх сторін.

Висновки. Впровадження локальних баз даних спрощує та прискорює процес збору і візуалізації інформації, значно підвищуючи ефективність обробки великих обсягів телеметрії, включаючи показники сенсорів газу. Застосування локальних архітектур у серверних рішеннях для моніторингу безпеки дозволяє досягти високого рівня автономності, кращого розподілу ресурсів та підвищеної надійності систем, знижуючи витрати і час на доставку аварійних повідомлень, а також забезпечуючи велику гнучкість при масштабуванні проектів без використання хмарних сервісів.

Список літератури

1. R. Buyya and A. V. Dastjerdi, *Internet of Things: Principles and Paradigms*, Elsevier, 2016, pp. 3-23.

2. W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin and X. Yang, *A Survey on the Edge Computing for the Internet of Things, IEEE Access*, vol. 6, 2018, pp. 6900-6919.

Костенюк А.В.
Студентка 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Мельник Г.М., кафедра КІ ЗУНУ

СТРУКТУРА МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ ЗБОРУ ЕКОЛОГІЧНИХ ПАРАМЕТРІВ

Вступ. Сучасні системи екологічного моніторингу відіграють важливу роль у контролі стану навколишнього середовища, особливо в умовах зростання урбанізації, збільшення кількості джерел забруднення та потреби в оперативному отриманні вимірювальних даних. Для таких задач доцільно використовувати автономні мікроконтролерні пристрої, які здатні періодично зчитувати параметри середовища, виконувати первинну обробку результатів і передавати їх на сервер або платформу Інтернету речей. Особливої актуальності набувають системи на основі контролера ESP32, оскільки він поєднує достатню обчислювальну продуктивність, підтримку бездротового зв'язку Wi-Fi, наявність цифрових та аналогових інтерфейсів, а також можливість використання режимів зниженого енергоспоживання. Це дозволяє створювати компактні сенсорні вузли для навчальних, побутових і прикладних задач екологічного моніторингу.

Постанова задачі. Метою роботи є розробка структури мікроконтролерної системи збору екологічних параметрів на основі контролера ESP32, яка забезпечує вимірювання основних параметрів навколишнього середовища, їх первинну обробку та передавання даних для подальшого аналізу.

Об'єктом дослідження є процес вимірювання, первинної обробки та передавання екологічних параметрів навколишнього середовища за допомогою мікроконтролерної системи.

Предметом дослідження виступають апаратно-програмні засоби побудови автономного сенсорного вузла на основі ESP32, призначеного для збору температури, вологості, атмосферного тиску, освітленості та показників якості повітря.

Основний матеріал. Структура мікроконтролерної системи збору екологічних параметрів побудована за модульним принципом. До її складу входять сенсорний модуль, мікроконтролерний модуль, модуль живлення, модуль передавання даних і серверна частина для збереження та візуалізації результатів. Центральним елементом системи є ESP32, який виконує функції керування сенсорами, обробки виміряних значень, формування телеметричного повідомлення та організації передавання даних у мережу.

Сенсорний модуль призначений для вимірювання параметрів навколишнього середовища. Для визначення температури та вологості може використовуватися цифровий датчик DHT11 або DHT22. Атмосферний тиск вимірюється за допомогою сенсора BMP280, який працює через інтерфейс I2C. Для контролю рівня освітленості доцільно застосувати модуль BH1750, що також використовує цифровий інтерфейс I2C і дозволяє отримувати значення освітленості в люксах. Для виявлення наявності газів або диму може бути використаний сенсор MQ-2, який формує аналоговий сигнал, що зчитується через вхід ADC контролера ESP32.

Мікроконтролерний модуль виконує роль центрального вузла керування. На першому етапі роботи ESP32 ініціалізує сенсори, перевіряє доступність інтерфейсів і встановлює з'єднання з мережею Wi-Fi. Після цього контролер послідовно зчитує дані з цифрових і аналогових вимірювальних каналів, перевіряє коректність значень, за потреби виконує усереднення або просту фільтрацію та формує набір телеметричних параметрів. Такий підхід дозволяє зменшити вплив випадкових похибок і підготувати дані до передавання на сервер.

Важливою частиною структури є організація обміну даними. Після зчитування параметрів система формує повідомлення у форматі JSON, наприклад із полями temperature, humidity, pressure, light, gas та battery. Далі повідомлення передається через Wi-Fi на серверну платформу або IoT-сервіс, наприклад ThingsBoard. Серверна частина забезпечує приймання телеметрії, збереження даних, побудову графіків і відображення поточного стану середовища на інформаційній панелі.

Окрему увагу в структурі системи приділено енергоспоживанню. Оскільки сенсорний вузол може працювати автономно, доцільно використовувати акумуляторне живлення та режим глибокого сну ESP32. У такому режимі контролер більшу частину часу перебуває у стані зниженого споживання, а активується лише для виконання циклу вимірювання і передавання даних. Це дозволяє збільшити тривалість автономної роботи системи без суттєвого ускладнення апаратної частини.

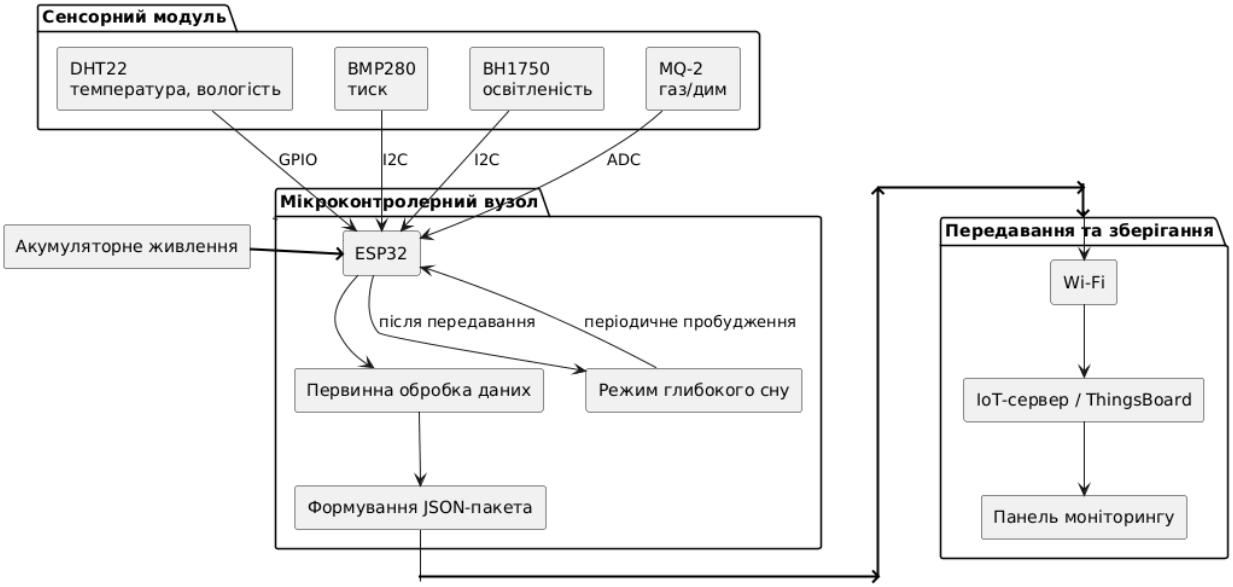


Рисунок 1 - Структура мікроконтролерної системи збору екологічних параметрів

Алгоритм роботи системи можна подати у вигляді послідовності таких етапів: пробудження контролера, ініціалізація сенсорів, зчитування екологічних параметрів, перевірка та первинна обробка даних, формування JSON-пакета, передавання даних на сервер, отримання підтвердження передавання та перехід у режим сну. Така послідовність є простою для реалізації в середовищі Arduino IDE та може бути адаптована під різні набори сенсорів.

Перевагою запропонованої структури є її масштабованість. За потреби до системи можна додати сенсори пилу, шуму, ультрафіолетового випромінювання або модуль бездротового зв'язку LoRa для передавання даних на більшу відстань. Модульний підхід також спрощує тестування, оскільки кожен сенсор можна перевірити окремо, а потім інтегрувати в загальний програмний код.

Висновки. Розглянута структура мікроконтролерної системи збору екологічних параметрів на основі ESP32 забезпечує поєднання вимірювальних, обчислювальних і комунікаційних функцій в одному автономному сенсорному вузлі. Використання цифрових сенсорів для температури, вологості, тиску й освітленості, а також аналогового газового сенсора дозволяє отримувати комплексну інформацію про стан навколишнього середовища.

Запропонована структура є придатною для реалізації навчального прототипу екологічного моніторингу, а також може бути основою для подальшого розвитку системи в напрямі підвищення автономності, додавання нових сенсорів, використання сонячного живлення та розширення способів передавання даних. Практичне значення роботи полягає у створенні гнучкої апаратно-програмної основи для збору, обробки та візуалізації екологічних параметрів у системах Інтернету речей.

Список літератури

1. Espressif Systems. ESP32 Series Datasheet. Espressif Systems, 2024.
2. Aosong Electronics. DHT11/DHT22 Temperature and Humidity Sensor Datasheet.
3. Bosch Sensortec. BMP280 Digital Pressure Sensor Datasheet.
4. ROHM Semiconductor. BH1750FVI Digital 16bit Serial Output Type Ambient Light Sensor IC Datasheet.

Рожанський О.О., Дикунець В.В.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Мельник Г.М., кафедра КІ ЗУНУ

ЗБІР І ПЕРЕДАВАННЯ ТЕЛЕМЕТРИЧНИХ ДАНИХ У РОЗПОДІЛЕНИХ СИСТЕМАХ МОНІТОРИНГУ

Вступ. Розподілені системи моніторингу є важливою складовою сучасних кіберфізичних систем, промислової автоматизації, енергетики, транспорту та Інтернету речей. Їх основним завданням є періодичне отримання даних від просторово рознесених об’єктів, первинна обробка виміряних параметрів і передавання результатів до серверної або хмарної інфраструктури для подальшого аналізу. Особливого значення набувають мікропроцесорні сенсорні вузли, здатні працювати автономно, виконувати вимірювання електричних параметрів, формувати телеметричні повідомлення та передавати їх через бездротові або дротові канали зв’язку.

Актуальність теми зумовлена потребою у створенні недорогих, енергоефективних і масштабованих засобів моніторингу розподілених об’єктів. У таких системах важливо забезпечити не лише точність вимірювання, а й надійність передавання даних, контроль втрат пакетів, ідентифікацію вузлів, можливість розширення мережі та інтеграцію з IoT-платформами. Для цього доцільно використовувати мікроконтролери, цифрові сенсори, модулі бездротового зв’язку та стандартні протоколи обміну телеметрією, зокрема MQTT, HTTP або LoRaWAN [1–3].

Постанова задачі. Метою роботи є дослідження принципів збору, первинної обробки та передавання телеметричних даних у розподілених системах моніторингу, а також обґрунтування структури мікропроцесорного сенсорного вузла для інтеграції з мережею Інтернету речей.

Об’єкт дослідження: процес вимірювання, первинної обробки та передавання телеметричних даних у мікропроцесорних системах моніторингу розподілених об’єктів.

Предмет дослідження: апаратно-програмні засоби побудови мікропроцесорного сенсорного вузла для вимірювання електричних параметрів, формування телеметричних повідомлень і передавання даних у мережу Інтернету речей.

Для досягнення поставленої мети необхідно розв’язати такі завдання: проаналізувати архітектуру розподіленої системи моніторингу; визначити склад мікропроцесорного сенсорного вузла; описати етапи вимірювання та первинної обробки електричних параметрів; запропонувати структуру телеметричного повідомлення; обґрунтувати спосіб передавання даних до IoT-сервера або шлюзу.

Основний матеріал. Розподілена система моніторингу складається з набору сенсорних вузлів, комунікаційного каналу, шлюзу, бази даних і користувацького інтерфейсу. Сенсорний вузол виконує функції збору вимірювальної інформації. У випадку моніторингу електричних параметрів до складу вузла можуть входити датчики напруги, струму, потужності, температури електронних компонентів, а також модуль контролю напруги живлення.

Апаратна частина вузла побудована на базі мікроконтролера з підтримкою інтерфейсів I2C, SPI, UART. Для вимірювання електричних параметрів доцільно застосовувати спеціалізовані сенсорні модулі, які забезпечують перетворення аналогових сигналів у цифрову форму. Для вимірювання струму та напруги можуть використовуватися модулі на основі вимірювальних мікросхем або шунтових перетворювачів.

Первинна обробка даних на рівні сенсорного вузла передбачає фільтрацію випадкових відхилень, усереднення серії вимірювань, перевірку граничних значень, обчислення похідних параметрів і підготовку даних до передавання. Для зменшення впливу шумів може використовуватися ковзне середнє або медіанна фільтрація. Середнє значення виміряного параметра можна визначити за формулою:

$$x_{сер} = (x_1 + x_2 + ... + x_n) / n,$$

де x_1, x_2, ..., x_n — окремі результати вимірювання, n — кількість вимірювань. Для електричних параметрів додатково можуть обчислюватися потужність, мінімальне та максимальне значення, середнє значення за інтервал часу, а також ознака перевищення допустимого порогу.

Важливим етапом роботи вузла є формування телеметричного повідомлення. Таке повідомлення повинно містити ідентифікатор вузла, часову мітку, набір виміряних параметрів, службові дані про стан живлення та, за потреби, параметри якості зв'язку. Для передавання в IoT-платформи зручно використовувати JSON-структуру такого вигляду:

```
{
"node_id": "node_01",
"timestamp": "2026-06-12T12:00:00",
"voltage": 12.4,
"current": 1.8,
"power": 22.3,
"battery": 86,
"rssi": -92
}
```

Перевагою такого формату є простота інтеграції з MQTT-брокером, сервером баз даних або платформами моніторингу. Водночас для мереж із низькою пропускною здатністю, наприклад LoRa, доцільно використовувати компактніший формат передавання, у якому числові значення кодуються у скороченому вигляді або передаються як бінарний пакет.

Передавання телеметричних даних може здійснюватися різними способами залежно від умов експлуатації. Для локальних систем із наявною інфраструктурою доцільним є використання Wi-Fi та MQTT, що забезпечує просту інтеграцію з серверними платформами. Для розподілених об'єктів, розміщених на значній відстані, доцільно застосовувати LoRa або LoRaWAN, оскільки ці технології орієнтовані на передавання невеликих обсягів даних на великі відстані з низьким енергоспоживанням [2,3]. У такій архітектурі сенсорний вузол передає пакет до шлюзу, шлюз пересилає дані до MQTT-брокера або серверної платформи.

Алгоритм роботи мікропроцесорного вузла можна подати як послідовність таких етапів: ініціалізація мікроконтролера та сенсорів; перевірка стану живлення; зчитування електричних параметрів; фільтрація та усереднення даних; формування телеметричного повідомлення; передавання пакета через вибраний канал зв'язку; очікування підтвердження або запис статусу передавання; перехід у режим енергозбереження до наступного циклу вимірювання. Такий підхід дозволяє зменшити споживання енергії та забезпечити стабільну роботу автономного вузла.

Практичне значення полягає у можливості побудови універсального сенсорного вузла для моніторингу електричних параметрів у розподілених об'єктах. Такий вузол може застосовуватися для контролю стану автономних джерел живлення, електричних мереж малих об'єктів, транспортних систем або елементів інфраструктури «розумного» міста.

Висновки. У роботі розглянуто підхід до збору і передавання телеметричних даних у розподілених системах моніторингу. Визначено, що ефективність таких систем залежить від узгодженої роботи сенсорного модуля, мікроконтролера, програмних алгоритмів первинної обробки та комунікаційного каналу. Запропоновано структуру мікропроцесорного сенсорного вузла, який забезпечує вимірювання електричних параметрів, попередню фільтрацію даних, формування телеметричного повідомлення та передавання інформації до IoT-інфраструктури.

Подальшим напрямом дослідження є практична реалізація прототипу сенсорного вузла, експериментальне вимірювання точності збору електричних параметрів, оцінка стабільності передавання телеметрії та дослідження енергоспоживання вузла в різних режимах роботи.

Список літератури

1. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems. 2013. Vol. 29, No. 7. P. 1645–1660.

2. Mekki K., Bajic E., Chaxel F., Meyer F. A comparative study of LPWAN technologies for large-scale IoT deployment. ICT Express. 2019. Vol. 5, No. 1. P. 1–7.

3. LoRa Alliance. LoRaWAN 1.0.4 Specification. Fremont: LoRa Alliance, 2020. 101 p.

Демедюк Т.В.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Загородня Д.І., кафедра ІОСУ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ АДАПТИВНОЇ СКЛАДНОСТІ ГЕЙМПЛЕЮ В 2D-ІГРАХ НА UNITY

Вступ. Сучасна індустрія відеоігор активно використовує інтелектуальні механізми адаптації ігрового процесу для підвищення залученості та задоволеності користувачів. Одним із таких механізмів є динамічне налаштування складності (Dynamic Difficulty Adjustment, DDA), яке дозволяє автоматично змінювати параметри гри відповідно до рівня майстерності гравця. Використання адаптивних систем складності сприяє підтриманню балансу між викликом та можливостями користувача, що позитивно впливає на ігровий досвід, мотивацію та тривалість взаємодії з грою. Особливо актуальним є впровадження таких підходів у динамічних іграх жанру top-down shooter, де складність ігрового процесу суттєво впливає на рівень зацікавленості гравця [1–3].

Постанова задачі. Метою роботи є розробка програмного модуля адаптивної складності для двовимірної гри жанру top-down shooter на платформі Unity, який забезпечує автоматичне коригування параметрів геймплею відповідно до поточних показників ефективності гравця. Об'єктом дослідження є процес динамічного балансування складності у комп'ютерних іграх. Предметом дослідження виступають методи оцінювання продуктивності гравця та алгоритми адаптивної зміни параметрів ігрового середовища на основі отриманих показників.

Основний матеріал. У процесі дослідження було проведено аналіз існуючих підходів до реалізації систем адаптивної складності в комп'ютерних іграх. Розглянуто чотири основні групи методів: системи зі статичним вибором рівня складності, правилові системи, підходи на основі машинного навчання та метрично-базовані методи. Порівняльний аналіз показав, що метрично-базований підхід поєднує достатню гнучкість, відносну простоту реалізації та відсутність потреби у попередньо підготовлених навчальних даних, що робить його найбільш придатним для використання в ігрових проєктах на платформі Unity [4].

Для реалізації програмного модуля було спроєктовано модульну архітектуру, що складається з декількох взаємопов'язаних компонентів. Центральним елементом системи є модуль збору телеметричних даних, який фіксує ключові ігрові події та передає їх до підсистеми оцінювання продуктивності гравця. На основі отриманих даних здійснюється розрахунок показників ефективності та визначення поточного рівня складності гри. Архітектура побудована за принципом слабкого зв'язування компонентів через механізм подій, що забезпечує можливість подальшого розширення функціоналу без значних змін у структурі програмного забезпечення.

Для оцінювання продуктивності гравця використовуються три основні показники:

– точність стрільби (Accuracy);

– кількість знищених супротивників за хвилину (Kills Per Minute, KPM);

– рівень отриманих ушкоджень за хвилину (Damage Per Minute, DPM).

На основі зазначених показників обчислюється інтегральний індекс складності Difficulty Index (DI), який відображає рівень успішності гравця у поточній ігровій ситуації. Для забезпечення стабільності оцінювання використовується механізм експоненціального ковзного середнього, що дозволяє згладжувати короткочасні коливання показників та уникати різких змін складності внаслідок випадкових помилок користувача [5].

Залежно від значення індексу DI система автоматично обирає один із трьох режимів функціонування:

– Assist – режим спрощення гри;

– Neutral – базовий режим;

– Challenge – режим підвищеної складності.

Перемикання між режимами здійснюється на основі визначених порогових значень індексу складності. Для запобігання частому перемиканню режимів реалізовано механізм

часової затримки (cooldown), який забезпечує плавність адаптації. Додатково передбачено механізм автоматичного зниження складності після загибелі персонажа, що дозволяє гравцю швидше відновити контроль над ситуацією та продовжити проходження гри.

Розроблений модуль реалізовано у вигляді п'яти MonoBehaviour-компонентів, розміщених на об'єкті _Systems: TelemetryManager (збір подій через патерн Observer), EvaluationEngine (обчислення метрик і DI), StrategyManager (класифікація DI в режими Assist / Neutral / Challenge), ParameterAdjuster (застосування параметрів) та DebugDdaUI (відображення стану під час тестування). Однонаправлений потік даних і слабке зв'язування компонентів через C#-події забезпечують модульність системи.

Для оцінки продуктивності гравця використовуються три нормалізовані метрики: точність стрільби (Accuracy), кількість знищених ворогів за хвилину (KPM) та отримана шкода за хвилину (DPM). Вони об'єднуються в інтегральний показник Difficulty Index (DI) за формулою [4]:

$$DI = 0.25 \times Accuracy + 0.35 \times SurvivabilityScore + 0.40 \times AggressionScore,$$

де SurvivabilityScore = 1 / (1 + DPM / 80) - нормалізований показник виживання; AggressionScore = KPM / 12 - нормалізований показник агресивності.

Значення DI знаходиться в діапазоні [0; 1], де значення, близьке до 1, означає, що гравець легко справляється з поточною складністю. Для стабілізації показника застосовується експоненціальне ковзне середнє (EMA) з коефіцієнтом $\alpha = 0.3$, що унеможливлює різкі коливання DI при короткочасних помилках гравця [6].

StrategyManager порівнює поточне DI з двома порогами (diMin = 0.35, diMax = 0.65) і перемикає один із трьох режимів: Assist (складність знижується), Neutral (базові параметри), Challenge (складність підвищується). Механізм cooldown тривалістю 1 секунда запобігає нестабільному мерехтінню режимів при коливанні DI поблизу порогу. При загибелі гравця спрацьовує Death Reset - DI примусово знижується до 0.2 через прискорене EMA-зміщення, що гарантує миттєве полегшення.

Усі параметри модуля (порогові значення, коефіцієнт α, діапазони ігрових параметрів для кожного режиму) налаштовуються через інспектор Unity без модифікації коду завдяки атрибутам [Header] і [Range]. Це забезпечує доступність інструменту для розробників із різним рівнем технічної підготовки [5].

Висновки. Розроблений програмний модуль реалізує метрично-базований підхід до DDA і усуває ключові недоліки існуючих рішень: забезпечує агрегацію кількох метрик із ваговими коефіцієнтами, плавні переходи між рівнями складності через лінійну інтерполяцію та повну модульність архітектури. Модуль може бути підключений до будь-якої 2D-гри на Unity без суттєвих змін у структурі проєкту. Практична апробація підтвердила ефективність системи: у режимі Challenge гра відчувалась значно інтенсивнішою, у режимі Assist - давала достатньо часу для відновлення, при цьому переходи між режимами залишались непомітними для гравця. Перспективою подальшого розвитку є розширення набору метрик та дослідження можливості інтеграції легкої ML-моделі для точнішого прогнозування оптимальної складності.

Список літератури

1. Csikszentmihalyi M. Flow: The Psychology of Optimal Experience. New York: Harper & Row, 1990. 303 p.

2. Silva W., Silva A., Chaimowicz L. Dynamic difficulty adjustment in video games: A review. Proceedings of the Brazilian Symposium on Games and Digital Entertainment (SBGames). 2017. P. 101–110.

3. Zhaowei M., Xiaoming Y. Heuristic Methods for Dynamic Difficulty Adjustment in 2D Shooter Games. IEEE Transactions on Games. 2022. Vol. 14, No. 3. P. 412–421.

4. Xie G. Exponential Moving Average Filter for Game Metrics Smoothing. Journal of Game Development Research. 2021. Vol. 8, No. 1. P. 89–96.

5. Unity Technologies. Unity User Manual (Version 6). URL: https://docs.unity3d.com/6000.0/Documentation/Manual/ .

6. Nystrom R. Game Programming Patterns. Genever Benning, 2014. 354 p.

Гуска А.А.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Загородня Д.І., кафедра ІОСУ ЗУНУ

ВЕБЗАСТОСУНОК ІНТЕРНЕТ-МАГАЗИНУ ЦИФРОВОЇ ТЕХНІКИ З ФАСЕТНИМ ПОШУКОМ ЗА ТЕХНІЧНИМИ ХАРАКТЕРИСТИКАМИ

Вступ. У сучасних умовах стрімкого розвитку електронної комерції ефективна навігація в товарних каталогах цифрової техніки набуває критичного значення. Сегмент електроніки характеризується високою розмірністю атрибутивного простору: кожен товар описується десятками унікальних технічних параметрів — діагоналлю та типом матриці екрана, поколінням процесора, обсягом оперативної та постійної пам'яті, ємністю акумулятора, частотою оновлення тощо. Класичний текстовий пошук та ієрархічна навігація за категоріями виявляються неспроможними задовольнити запити користувачів, які прагнуть звузити пропозиції за специфічним набором параметрів. Згідно зі статистичними даними, у 2024–2025 роках понад 78% інтернет-користувачів європейського регіону регулярно купують онлайн, а сектор цифрової електроніки лише у Польщі згенерував 5,86 мільярда доларів США доходу [1]. У таких умовах якість пошукової підсистеми стає ключовим чинником конкурентоспроможності інтернет-магазину.

Стандартом де-факто для електронної комерції з широкими каталогами стали системи фасетного пошуку (англ. faceted search). Вони дозволяють користувачеві паралельно звужувати вибірку за множинними ортогональними критеріями та одночасно агрегують у режимі реального часу кількість товарів за кожним фасетом ще до застосування відповідного фільтра [2]. Реалізація таких систем потребує продуманої архітектури зберігання слабоструктурованих атрибутів, оптимізованих SQL-запитів із мілісекундним відгуком та узгодженого UX/UI. Метою цього дослідження є розробка вебзастосунку інтернет-магазину цифрової техніки з ефективною підсистемою фасетного пошуку та фільтрації за технічними характеристиками.

Постановка задачі. Проведений аналіз існуючих платформ електронної комерції — Magento, Shopify, WooCommerce — показав, що готові SaaS-рішення суттєво обмежують свободу проєктування бази даних і не дозволяють реалізовувати специфічну логіку ранжування та формування фасетів. Magento (Adobe Commerce) надає підтримку складних каталогів через модель EAV (англ. Entity-Attribute-Value), однак ця модель є класичним антипатерном реляційних баз даних: для одночасної фільтрації за n атрибутами вимагається n операцій самоз'єднання, що в каталогах із сотнями тисяч записів призводить до катастрофічної деградації продуктивності. Експериментальні тести демонструють, що типовий запит за перетином атрибутів у моделі JSONB з GIN-індексуванням виконується за 181 мс, тоді як аналогічний запит до структури EAV потребує понад 7000 мс — більш ніж у 39 разів повільніше [3].

Об'єктом дослідження є процес пошуку та фасетної фільтрації товарів в інтернет-магазині цифрової техніки. Предметом дослідження є методи, моделі та програмні засоби, що реалізують фасетний пошук, зберігання динамічних технічних атрибутів та оптимізацію продуктивності SQL-запитів. Метою роботи є проєктування та реалізація вебзастосунку інтернет-магазину цифрової техніки, який забезпечує мілісекундний відгук на запити фасетної фільтрації за довільним набором технічних характеристик.

Для досягнення поставленої мети необхідно розв'язати такі задачі: спроєктувати реляційну схему бази даних із гібридним зберіганням типізованих полів та слабоструктурованих характеристик у форматі JSONB; реалізувати RESTful API з ендпоінтами фасетної агрегації; розробити клієнтську частину з динамічною панеллю фільтрів, чутливою до обраної категорії; забезпечити коректне обчислення лічильників фасетів, при якому вибір значення в одному вимірі не приховує інші доступні значення в тому ж вимірі.

Основний матеріал. Архітектурно розроблений вебзастосунок Volt побудовано за класичною триланковою схемою (рисунок 1). Клієнтська частина реалізована на React 18 із

застосуванням Vite як інструменту збирання, Tailwind CSS — для стилізації, React Router — для маршрутизації. Серверна частина побудована на Node.js і фреймворку Express, який обслуговує RESTful-ендпоінти [4]. Як систему управління базою даних обрано PostgreSQL, що нативно підтримує тип JSONB та інвертовані індекси GIN, необхідні для ефективної фасетної агрегації.

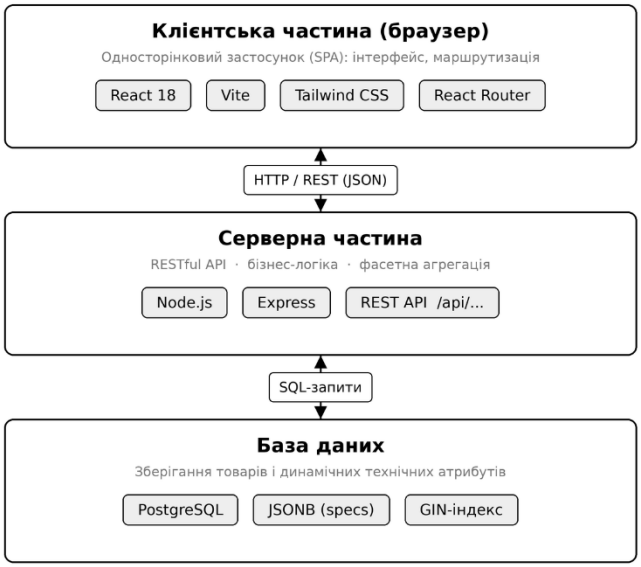


Рисунок 1 – Триланкова архітектура вебзастосунку інтернет-магазину

Центральним архітектурним рішенням є гібридна модель зберігання атрибутів товару. Типізовані поля, спільні для всіх товарів (назва, slug, артикул, ціна, залишок, ідентифікатори бренду та категорії, рейтинг), розміщено в окремих реляційних колонках таблиці products. Натомість специфічні технічні характеристики, набір яких відрізняється для кожної категорії (наприклад, обсяг RAM та роздільна здатність камери для смартфонів; діагональ і тип матриці — для телевізорів), зберігаються в колонці specs типу JSONB. Над цією колонкою побудовано GIN-індекс, що забезпечує ефективний пошук за наявністю ключа та значенням безпосередньо в бінарному поданні JSONB [5]. Схема атрибутів кожної категорії визначається в окремій таблиці attribute_definitions, яка зберігає метадані: технічну назву, відображувану назву, тип даних (number, enum, boolean, string), одиницю виміру та опції перелічень.

Серверна логіка пошуку реалізована в ендпоінті GET /api/products/filters, який повертає набір фасетів для поточної категорії з лічильниками. Ключовим аспектом є коректний підрахунок цих лічильників. Якщо для кожного виміру застосовувати повний поточний набір фільтрів, значення в обраному вимірі стають недоступними після першого вибору, що блокує користувачеві подальшу зміну рішення. Для розв’язання цієї задачі для кожного виміру v обчислюється лічильник без власного обмеження, тобто з виключенням f_v з множини активних фільтрів. Формально, для каталогу товарів P та активних фільтрів F = {f_1, f_2, …, f_n} кількість товарів для значення x фасета v визначається як:

$$count_v(x) = |\{\, p \in P : (\forall f_i \in F \setminus \{f_v\})\, f_i(p) \wedge p.specs[v] = x \,\}| \qquad (1)$$

Такий підхід гарантує, що користувач завжди бачить альтернативні значення в межах обраного виміру та може коректно змінити свій вибір без втрати контексту запиту.

Клієнтська частина реалізує панель фільтрів, склад якої завантажується динамічно за обраною категорією. Числові атрибути (RAM, ємність акумулятора, частота оновлення) подаються у вигляді чипів зі значеннями; перелічувані (бренд, колір, тип матриці) — у вигляді багатовибірних чекбоксів; булеві (NFC, підтримка 5G, бездротова зарядка) — як одиночні перемикачі. Ціновий діапазон керується двозональним повзунком із прив’язкою до полів ручного введення. Зміни фільтрів відтерміновано накопичуються в локальному стані (draft) і застосовуються до URL-параметрів лише після натискання кнопки «Показати N товарів», що мінімізує кількість мережевих запитів і забезпечує плавність взаємодії.

Окрім підсистеми фасетного пошуку, у роботі реалізовано повний цикл функцій електронної комерції: реєстрація користувачів із підтвердженням пошти через email-токен у шляху URL (для уникнення обрізання query-параметрів поштовими провайдерами), відновлення пароля, оформлення замовлень із трьома варіантами доставки (кур'єр, поштова служба, самовивіз), система бонусних балів з нарахуванням 1,5% від суми замовлення, можливість залишити відгук із оцінкою та опційним текстом, керування поверненнями. Адміністративна панель реалізує CRUD-операції над товарами, категоріями та брендами; форма створення товару динамічно формує інпути специфічних характеристик на основі attribute_definitions поточної категорії, авто-генерує числовий артикул через окрему PostgreSQL-послідовність та забезпечує завантаження зображень через multipart-форму з валідацією MIME-типу та обмеженням розміру.

Висновки. У ході виконаного дослідження спроєктовано та реалізовано повнофункціональний вебзастосунок інтернет-магазину цифрової техніки з підсистемою фасетного пошуку та фільтрації за технічними характеристиками. Обрано гібридну модель зберігання атрибутів на основі PostgreSQL JSONB з GIN-індексуванням, яка, згідно з опублікованими порівняльними тестами, забезпечує приблизно в 40 разів кращу продуктивність порівняно з моделлю EAV. Запропоновано та програмно реалізовано алгоритм агрегації фасетних лічильників із виключенням власного виміру, що гарантує коректну поведінку інтерфейсу при послідовному уточненні запиту користувачем. Архітектурні рішення апробовано на каталозі з п'ятьма категоріями (смартфони, ноутбуки, телевізори, аудіотехніка, аксесуари) та більш ніж 14 атрибутами на категорію, з підтримкою всього життєвого циклу замовлення — від кошика до бонусних нарахувань. Подальші напрями розвитку охоплюють інтеграцію алгоритмів машинного навчання для контекстно-залежного ранжування фасетів та оптимізацію клієнтської частини за метриками Core Web Vitals.

Список літератури

1. PolSenior2 Report. State of E-commerce in Poland: Growth, Trends, and Consumer Insights / Statista Market Forecast. 2025. URL: https://www.statista.com/outlook/emo/ecommerce/poland (дата звернення: 28.05.2026).

2. Wong D. Modern E-commerce Architecture: Building Scalable Online Stores. Manning Publications. 2023. 392 p.

3. Petković D. Comparison of JSON and XML Data Formats for PostgreSQL Database. Proceedings of the 44th International Convention MIPRO. 2021. P. 1546–1551.

4. Karau H. Best Practices for RESTful API Design. Sebastopol : O'Reilly Media. 2024. 256 p.

5. PostgreSQL 16 Documentation: Chapter 8 — Data Types, JSON Types [Електронний ресурс] / The PostgreSQL Global Development Group. – Режим доступу: https://www.postgresql.org/docs/16/datatype-json.html (дата звернення: 28.05.2026).

Копилов М.О.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Биковий П.Є., кафедра ІОСУ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ АВТОМАТИЗОВАНОГО ЗБОРУ ТА ОЦІНЮВАННЯ ВІДПОВІДЕЙ LLM-МОДЕЛЕЙ

Вступ. Стрімкий розвиток великих мовних моделей (Large Language Models, LLM) зумовив їх широке використання у програмуванні, освіті, бізнес-аналітиці та наукових дослідженнях. Водночас актуальною залишається проблема об'єктивного оцінювання якості відповідей різних моделей, оскільки результати їх роботи можуть відрізнятися за точністю, повнотою та релевантністю навіть для однакових запитів [1–4]. Проведення ручного тестування є трудомістким і не забезпечує необхідного рівня відтворюваності результатів, що обумовлює необхідність створення спеціалізованих програмних засобів автоматизованого збору та оцінювання відповідей LLM.

Постановка задачі. Метою роботи є розробка програмного модуля для автоматизованого збору, структурування та оцінювання відповідей великих мовних моделей на наборах тестових запитів. Об'єктом дослідження є процес взаємодії з великими мовними моделями та аналіз якості сформованих ними відповідей. Предметом дослідження виступають методи автоматизованого оцінювання відповідей LLM, алгоритми обробки природної мови та програмні засоби інтеграції з API мовних моделей.

Основний матеріал. У роботі проведено аналіз сучасних великих мовних моделей, архітектур Transformer та Retrieval-Augmented Generation (RAG), а також методів автоматизованого оцінювання якості текстових відповідей [1–5]. Особливу увагу приділено використанню класичних метрик BLEU, ROUGE, семантичної подібності на основі embeddings та підходу LLM-as-a-Judge для комплексного оцінювання результатів генерації [6, 7].

Загальна структура розробленого програмного модуля наведена на рисунку 1. Архітектура системи забезпечує повний цикл роботи з великими мовними моделями: від отримання тестових запитів та надсилання їх через API до автоматизованого аналізу отриманих відповідей і формування підсумкових оцінок. Використання модульного підходу забезпечує можливість підключення нових моделей та методів оцінювання без зміни загальної структури системи.

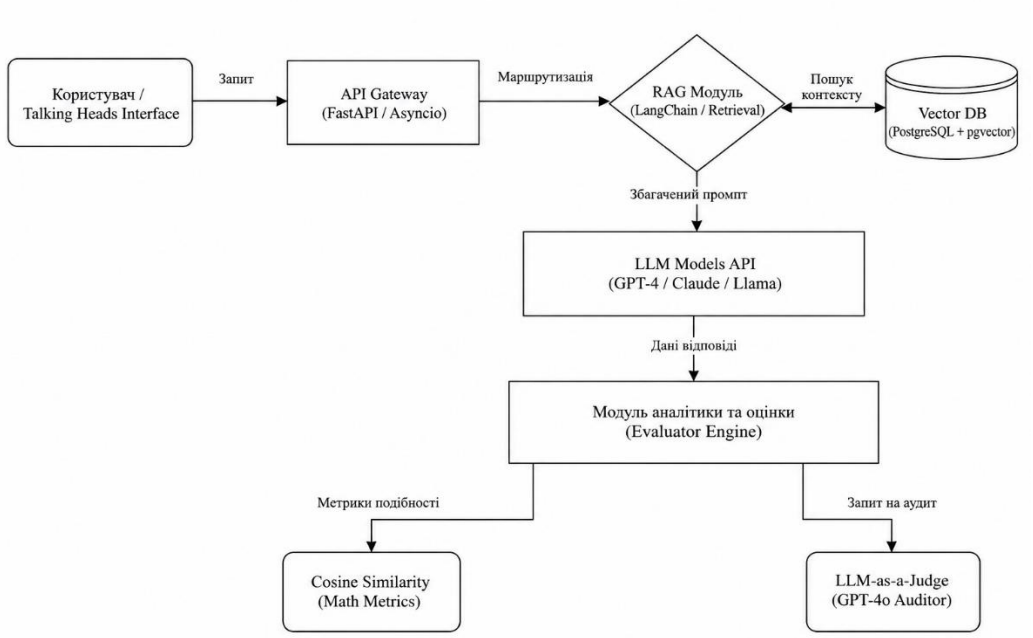


Рисунок 1 – Загальна структура програмного модуля автоматизованого збору та оцінювання відповідей LLM-моделей

Як показано на рисунку 1, користувацькі запити надходять через API Gateway до RAG-модуля, який виконує пошук релевантного контексту у векторній базі даних PostgreSQL з розширенням pgvector. Сформований запит передається до LLM-моделей, а отримані відповіді оцінюються за допомогою метрик семантичної подібності та підходу LLM-as-a-Judge.

Для реалізації програмного модуля використано мову програмування Python, сучасні засоби асинхронної взаємодії із зовнішніми API та систему керування базами даних PostgreSQL [8]. Програмне забезпечення підтримує автоматизований збір відповідей різних моделей, централізоване збереження результатів та виконання багатокритеріального аналізу якості відповідей.

Проведене експериментальне тестування підтвердило працездатність розробленого програмного модуля та можливість його використання для порівняльного аналізу сучасних мовних моделей. Отримані результати можуть бути використані під час вибору LLM для конкретних прикладних задач, а також у дослідженнях, пов’язаних із забезпеченням достовірності та якості генеративного штучного інтелекту.

Висновки. У результаті проведеного дослідження проаналізовано сучасні підходи до побудови та оцінювання великих мовних моделей. Розроблено архітектуру програмного модуля автоматизованого збору та оцінювання відповідей LLM, яка забезпечує інтеграцію з різними AI-сервісами, автоматизований аналіз результатів та підтримку масштабування системи. Запропоноване рішення може бути використане для проведення бенчмаркінгу мовних моделей, оцінювання якості генерації тексту та підтримки процесів вибору оптимальних моделей для практичного застосування.

Список літератури

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016.

2. Murphy K.P. Machine Learning: A Probabilistic Perspective. Cambridge : MIT Press, 2012.

3. Jurafsky D., Martin J.H. Speech and Language Processing. 3rd ed. Draft, 2024.

4. Vaswani A. et al. Attention Is All You Need // Advances in Neural Information Processing Systems (NeurIPS). 2017.

5. Lewis P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems (NeurIPS). 2020.

6. Papineni K. et al. BLEU: a Method for Automatic Evaluation of Machine Translation // Proceedings of ACL. 2002.

7. Zheng L. et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena // NeurIPS Datasets and Benchmarks Track. 2023.

8. PostgreSQL Global Development Group. PostgreSQL Documentation. URL: https://www.postgresql.org/docs/

Калюх Д.К.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Биковий П.Є., кафедра ІОСУ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ КЕРУВАННЯ НАВЧАЛЬНИМИ ДЕФЕКТАМИ ІНТЕРНЕТ-МАГАЗИНУ ДЛЯ ФОРМУВАННЯ НАВИЧОК ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ. Якість програмного забезпечення є одним із ключових факторів успішного функціонування сучасних інформаційних систем. Особливо важливу роль відіграє підготовка фахівців із забезпечення якості програмного забезпечення (QA), які повинні володіти практичними навичками виявлення та документування дефектів. Однією з актуальних проблем навчання тестуванню є відсутність реалістичних навчальних середовищ із керованими дефектами, які дозволяють відтворювати різноманітні помилки в умовах, максимально наближених до реальних проєктів [1–4].

Постановка задачі. Метою роботи є розробка програмного модуля керування навчальними дефектами інтернет-магазину на базі платформи Bagisto та фреймворку Laravel 12 для формування практичних навичок тестування програмного забезпечення. Об’єктом дослідження є процес підготовки QA-спеціалістів у контрольованому навчальному середовищі. Предметом дослідження виступають архітектурні та програмні засоби реалізації системи керування навчальними дефектами вебзастосунків електронної комерції.

Основний матеріал. У роботі проведено аналіз існуючих підходів до організації навчальних середовищ для тестування програмного забезпечення, зокрема навчальних стендів електронної комерції, платформ OWASP Juice Shop, систем управління дефектами Jira та сучасних платформ електронної комерції на базі Laravel [5–8]. Встановлено, що більшість існуючих рішень використовують статично закладені дефекти та не забезпечують можливості їх динамічного керування.

Для усунення виявлених недоліків розроблено програмний модуль BugManager, який реалізовано у вигляді окремого пакета для платформи Bagisto. Архітектура системи базується на використанні фреймворку Laravel 12 та механізмів Service Provider, Middleware і Eloquent ORM. Модуль забезпечує централізоване керування навчальними дефектами через адміністративну панель без внесення змін до основного коду інтернет-магазину.

До складу системи входять три категорії навчальних дефектів: UI/UX, форми та валідація, API/Backend. Загалом реалізовано дев’ять типів дефектів, які можуть динамічно активуватися або деактивуватися адміністратором під час проведення лабораторних занять. Такий підхід дозволяє формувати різні навчальні сценарії та забезпечує відтворюваність результатів тестування. Реалізація модуля дозволяє формувати індивідуальні навчальні сценарії для різних рівнів підготовки студентів та контролювати складність тестових завдань.

Загальну структуру програмного модуля наведено на рисунку 1. Система складається з клієнтського рівня (браузери студента та викладача), платформи Bagisto, пакета BugManager, який містить компоненти InjectBugMiddleware, BugController та модель Bug, а також бази даних MySQL. Модуль забезпечує динамічне ввімкнення та вимкнення навчальних дефектів і їх ін’єкцію у функціональні компоненти інтернет-магазину без внесення змін до вихідного коду ядра платформи Bagisto.

Керування станом дефектів здійснюється через модель Bug та контролер BugController, тоді як механізм InjectBugMiddleware виконує перехоплення HTTP-запитів і модифікацію відповідей відповідно до активованих сценаріїв навчальних дефектів.

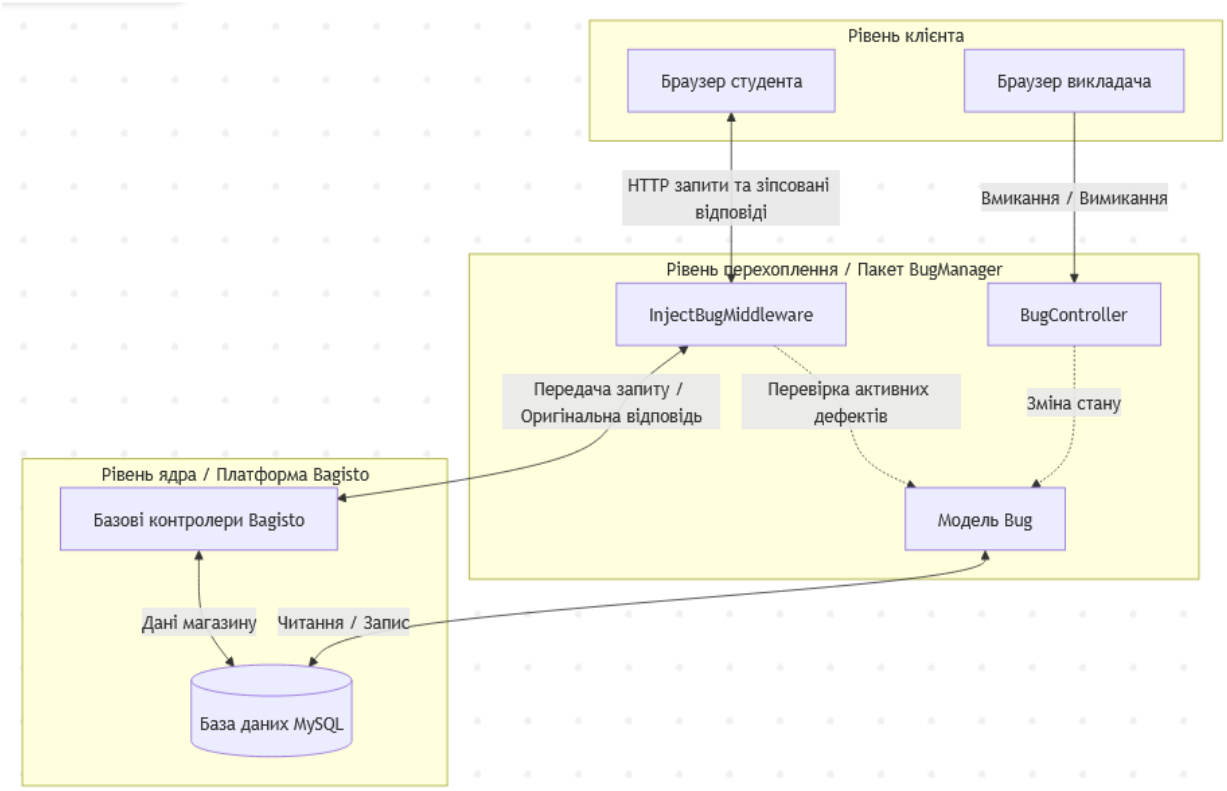


Рисунок 1 – Загальна структурна схема програмного модуля керування навчальними дефектами інтернет-магазину

Проведене тестування підтвердило коректність роботи механізмів керування дефектами та можливість їх використання для організації практичних занять із тестування програмного забезпечення. Розроблений модуль забезпечує гнучке формування навчальних сценаріїв та може бути інтегрований до освітніх програм підготовки QA-спеціалістів.

Висновки. У результаті дослідження проаналізовано сучасні підходи до організації навчальних середовищ для тестування програмного забезпечення. Розроблено програмний модуль BugManager для керування навчальними дефектами інтернет-магазину на базі Laravel 12 та Bagisto. Запропоноване рішення забезпечує динамічне керування дефектами різних категорій, підвищує ефективність формування практичних навичок тестування та може використовуватися як навчальна лабораторія для підготовки QA-спеціалістів.

Список літератури

1. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2019. 992 p.
2. Sommerville I. Software Engineering. 10th ed. Boston : Pearson Education, 2016. 816 p.
3. ISO/IEC/IEEE 29119-1:2022. Software and Systems Engineering — Software Testing — Part 1: General Concepts. Geneva : International Organization for Standardization, 2022.
4. Myers G. J., Sandler C., Badgett T. The Art of Software Testing. 3rd ed. Hoboken : John Wiley & Sons, 2011. 256 p.
5. OWASP Foundation. OWASP Juice Shop Documentation. URL: https://owasp.org/www-project-juice-shop/ (дата звернення: 15.02.2026).
6. Bagisto Team. Bagisto Documentation. URL: https://devdocs.bagisto.com (дата звернення: 15.02.2026).
7. Laravel LLC. Laravel 12 Documentation. URL: https://laravel.com/docs/12.x (дата звернення: 15.02.2026).
8. Atlassian. Jira Software Documentation. URL: https://www.atlassian.com/software/jira (дата звернення: 15.02.2026).

Гончарук К.С.
Студентка 4 курсу ФКІТ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ КЛАСИФІКАЦІЇ СТАНІВ ФІНАНСОВОГО РИНКУ З ВИКОРИСТАННЯМ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ

Вступ. Сучасні фінансові та криптовалютні ринки характеризуються високою динамічністю, значними обсягами даних та високим рівнем невизначеності. Аналіз ринкових процесів у режимі реального часу потребує використання інтелектуальних методів обробки даних, здатних автоматично виявляти закономірності та класифікувати поточний стан ринку. Застосування методів машинного навчання дозволяє підвищити об'єктивність аналізу та забезпечити підтримку прийняття торгових рішень на основі історичних і поточних даних фінансового ринку [1–4].

Постанова задачі. Метою роботи є розробка програмного модуля для автоматизованої класифікації станів фінансового ринку на основі даних криптовалютних бірж із використанням методів штучного інтелекту. Об'єктом дослідження є процес аналізу та класифікації станів фінансового ринку. Предметом дослідження виступають методи машинного навчання, алгоритми аналізу часових рядів та програмні засоби обробки фінансових даних.

Основний матеріал. У рамках дослідження проведено аналіз сучасних підходів до автоматизованого аналізу фінансових ринків та використання методів штучного інтелекту для підтримки прийняття рішень. Особливу увагу приділено концепції Smart Money Concepts, яка дозволяє формалізувати структуру ринку на основі аналізу ліквідності, зон накопичення, маніпуляцій та зламів структури [5].

Для забезпечення автоматизованого аналізу розроблено програмний модуль, архітектура якого включає підсистеми збору ринкових даних, формування простору ознак, класифікації станів ринку та генерації рекомендацій для користувача.

Архітектуру розробленого програмного модуля наведено на рисунку 1. Система реалізована за сервіс-орієнтованим підходом та включає модуль збору ринкових даних через API криптовалютних бірж, ETL-конвеєр обробки даних та обчислювальне ядро класифікації станів ринку на основі оптимізованого ансамблевого алгоритму градієнтного бустингу (XGBoost).

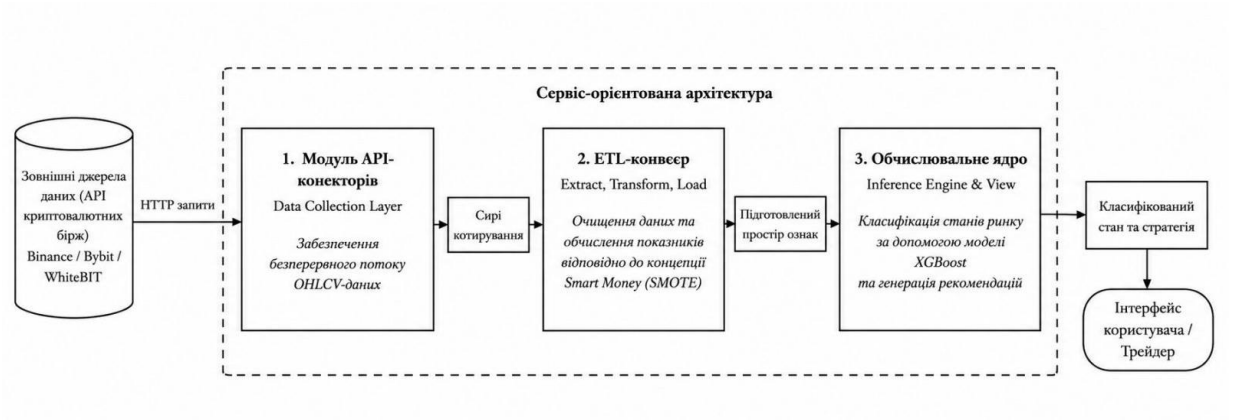


Рисунок 1 – Архітектура програмного модуля класифікації станів фінансового ринку

Як показано на рисунку 1, джерелом даних виступають API криптовалютних бірж, зокрема Binance, Bybit та WhiteBIT. Отримані котирування передаються до ETL-конвеєра, де виконується очищення даних, обробка пропусків та розрахунок показників відповідно до концепції Smart Money Concepts. На основі сформованого простору ознак обчислювальне ядро виконує класифікацію поточного стану ринку за допомогою моделі градієнтного бустингу XGBoost та формує рекомендації щодо торгової стратегії для користувача.

Одним із ключових елементів концепції Smart Money Concepts є цикл AMD (Accumulation–Manipulation–Distribution), який описує типову поведінку великого капіталу на фінансовому ринку. Схематичне представлення даного циклу наведено на рисунку 2.

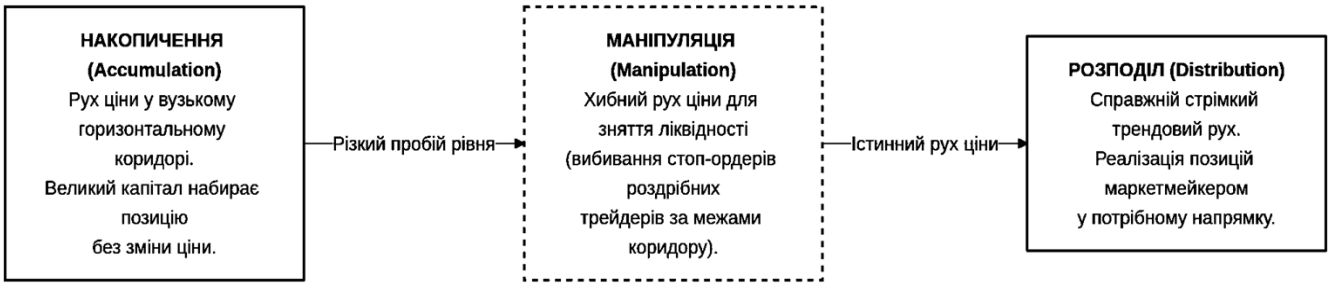


Рисунок 2 – Цикл Accumulation–Manipulation–Distribution (AMD) у концепції Smart Money Concepts

Цикл AMD використовується для формування частини ознак, що надалі подаються на вхід моделі машинного навчання для класифікації поточного стану ринку. Джерелом даних виступають API криптовалютних бірж Binance, Bybit та WhiteBIT, через які отримуються історичні та поточні ринкові дані. На основі цих даних формується набір ознак, що характеризують поточний стан ринку та включають показники структури тренду, ліквідності, зон цінової неефективності та поведінкових патернів ринку [6].

Для класифікації станів фінансового ринку використано оптимізований ансамблевий алгоритм градієнтного бустингу XGBoost [7]. Модель дозволяє автоматично відносити поточний стан ринку до одного з п’яти класів: накопичення, висхідний тренд, низхідний тренд, маніпуляція або злам структури. На основі результатів класифікації система формує аналітичні рекомендації щодо прийняття торгових рішень.

Програмний модуль реалізовано мовою Python із використанням бібліотек Pandas, NumPy, Scikit-learn, XGBoost, imblearn (для балансування даних) та SQLAlchemy. Для зберігання та обробки даних використано систему керування базами даних PostgreSQL [8]. Архітектура рішення забезпечує можливість автоматизованого збору, аналізу та класифікації фінансових даних у режимі реального часу.

Проведені експериментальні дослідження повністю підтвердили працездатність запропонованого підходу. Зокрема, застосування алгоритму SMOTE для усунення екстремального дисбалансу навчальної вибірки дозволило системі ефективно розпізнавати складні та рідкісні ринкові ситуації (маніпуляції ліквідністю, злами структури). Отримані результати доводять високу точність розробленої моделі та можуть бути використані як надійна основа для побудови сучасних інтелектуальних систем підтримки прийняття рішень у сфері фінансової аналітики.

Висновки. У результаті проведеного дослідження проаналізовано сучасні підходи до аналізу фінансових ринків та застосування методів штучного інтелекту для класифікації ринкових станів в умовах стохастичної невизначеності. Розроблено архітектуру програмного модуля, що забезпечує автоматизований збір, обробку та аналіз високочастотних фінансових даних. Запропоноване рішення поєднує оптимізовані ансамблеві методи машинного навчання, алгоритми подолання дисбалансу даних та концепцію Smart Money Concepts для визначення поточного стану ринку. Розроблений програмний модуль може бути успішно використаний як ядро інтелектуальних систем підтримки прийняття рішень (СППР) та управління ризиками на сучасних криптовалютних біржах.

Список літератури

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 800 p.

2. Murphy K.P. Machine Learning: A Probabilistic Perspective. Cambridge : MIT Press, 2012. 1067 p.

3. Aggarwal C.C. Neural Networks and Deep Learning. Cham : Springer, 2018. 497 p.
4. Tsay R.S. Analysis of Financial Time Series. 3rd ed. Hoboken : Wiley, 2010. 715 p.
5. Huddleston M.J. The Inner Circle Trader. Market Structure, Liquidity and Smart Money Concepts. 2022. URL: https://innercircletrader.net (дата звернення: 20.05.2026).
6. Binance Developers. Binance API Documentation. URL: https://developers.binance.com/docs
7. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). New York: ACM, 2016. P. 785–794
8. PostgreSQL Global Development Group. PostgreSQL Documentation. URL: https://www.postgresql.org/docs/

Сідлецький Т. А.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Загородня Д.І., кафедра ІОСУ ЗУНУ

ВИЯВЛЕННЯ ТА ВІДСТЕЖЕННЯ РУХОМИХ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ НА ВБУДОВАНІЙ ОБЧИСЛЮВАЛЬНІЙ ПЛАТФОРМІ BEAGLEBONE AI

Вступ. Сучасний розвиток систем комп'ютерного зору нерозривно пов'язаний із впровадженням технологій периферійних обчислень (Edge Computing), які забезпечують обробку даних безпосередньо на пристроях їх збору. Такий підхід є особливо актуальним для систем відеоспостереження, автономної навігації, інтелектуальних транспортних систем та мобільної робототехніки, де необхідно забезпечити мінімальні затримки передачі даних та високу швидкість прийняття рішень. Водночас використання сучасних моделей глибокого навчання на вбудованих платформах супроводжується значними обчислювальними труднощами через обмежені ресурси процесора, пам'яті та енергоспоживання. Тому актуальним є розроблення ефективних алгоритмів і програмних засобів, здатних забезпечити функціонування систем виявлення та відстеження об'єктів у режимі реального часу на гетерогенних обчислювальних платформах [1–4].

Постанова задачі. Метою роботи є розробка програмного модуля виявлення та відстеження рухомих об'єктів на платформі BeagleBone AI-64 із використанням апаратного прискорення та раціонального розподілу обчислювального навантаження між гетерогенними обчислювальними ядрами. Об'єктом дослідження є процес виявлення та відстеження рухомих об'єктів у відеопотоці в режимі реального часу. Предметом дослідження є алгоритми, методи та програмні засоби апаратного прискорення процесів виявлення та відстеження об'єктів на гетерогенній платформі BeagleBone AI-64.

Основний матеріал. У рамках дослідження проведено аналіз сучасних методів детекції та одночасного відстеження множини об'єктів у відеопотоках. Особливу увагу приділено архітектурним особливостям системи-на-кристалі TDA4VM, яка використовується у платформі BeagleBone AI-64 та містить ядра ARM Cortex-A72, цифрові сигнальні процесори DSP архітектури C7x і матричний прискорювач MMA [5].

Для забезпечення високої продуктивності розроблено багатопотокову архітектуру програмного модуля на основі конвеєрного шаблону проєктування (Pipeline Design Pattern). Архітектура складається з чотирьох взаємопов'язаних підсистем: захоплення відеоданих, детекції об'єктів, відстеження та візуалізації результатів. Взаємодія між підсистемами реалізована через потокобезпечні черги даних за принципом «виробник–споживач», що дозволяє ефективно розподіляти навантаження між апаратними компонентами платформи.

Загальну функціональну структуру розробленого програмного модуля наведено на рисунку 1. Система реалізована за конвеєрним принципом та включає етапи ініціалізації,

захоплення відеоданих, аналізу та детекції об’єктів, відстеження й ідентифікації об’єктів, а також візуалізації результатів. Підсистема детекції реалізує передобробку тензорів, нейромережевий інференс та NMS-фільтрацію, тоді як підсистема відстеження використовує екстракцію ознак, фільтр Калмана та Угорський алгоритм для підтримки траєкторій руху об’єктів.

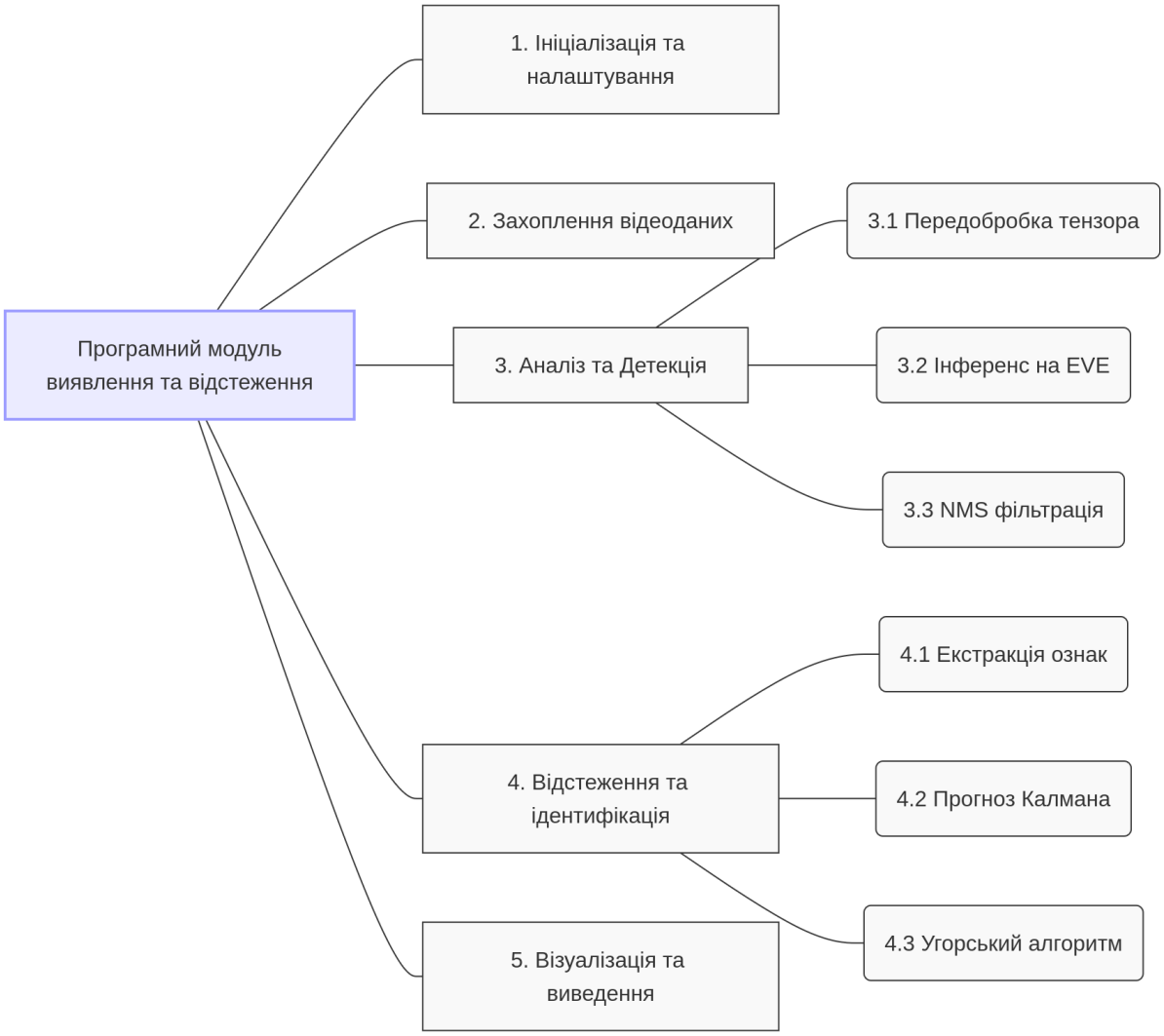


Рисунок 1 – Функціональна структура програмного модуля виявлення та відстеження рухомих об’єктів на платформі BeagleBone AI-64

Для детекції об’єктів використовується нейромережева модель YOLOv8 [1], адаптована до роботи на вбудованій платформі шляхом застосування посттренувального квантування до формату INT8 із використанням інструментарію TIDL [4]. Завдяки цьому значна частина обчислень виконується на прискорювачі C7x/MMA, що дозволяє зменшити навантаження на центральний процесор та підвищити швидкодію системи.

Підсистема відстеження реалізована на основі алгоритму DeepSORT [3], який поєднує прогнозування положення об’єктів за допомогою фільтра Калмана та асоціацію виявлень із наявними треками за допомогою Угорського алгоритму. Для підвищення стійкості відстеження використовуються візуальні дескриптори, отримані з нейромережевої моделі повторної ідентифікації (Re-ID). Формування матриці вартості здійснюється на основі поєднання просторових характеристик та косинусної відстані між векторами ознак, що забезпечує стабільне збереження ідентифікаторів об’єктів навіть в умовах часткових перекриттів та тимчасового зникнення з поля зору камери.

Загальна структура програмного модуля включає такі функціональні блоки:

- блок захоплення відео (забезпечує отримання відеопотоку від камери або відеофайлу та передачу кадрів до конвеєра обробки);
- блок детекції об’єктів (виконує попередню обробку кадрів, нейромережеву детекцію та формування набору обмежувальних рамок);
- блок відстеження об’єктів (реалізує екстракцію ознак, прогнозування траєкторій руху, асоціацію даних та підтримку життєвого циклу треків);
- блок візуалізації (забезпечує відображення результатів роботи системи у вигляді графічних позначок, ідентифікаторів об’єктів та траєкторій їх переміщення).

Інформаційна взаємодія між підсистемами реалізується через передачу структурованих даних, які містять відеокадри, результати детекції та параметри активних треків. Для забезпечення коректної синхронізації потоків використовуються механізми м’ютексів та умовних змінних, а також стратегія відкидання застарілих кадрів у разі перевантаження системи.

Програмний модуль реалізовано мовою C++ із використанням операційної системи Linux, бібліотеки OpenCV та програмного інтерфейсу TIDL для виконання нейромережевого інференсу на апаратних прискорювачах платформи.

Проведені експериментальні дослідження підтвердили працездатність запропонованого підходу та можливість виконання нейромережевого інференсу і відстеження об’єктів у режимі реального часу на платформі BeagleBone AI-64.

Висновки. У результаті проведеного дослідження виконано аналіз сучасних методів виявлення та відстеження рухомих об’єктів, а також особливостей їх реалізації на вбудованих Edge AI-платформах. Розроблено архітектуру гібридного програмного модуля, яка забезпечує ефективне використання гетерогенних обчислювальних ресурсів платформи BeagleBone AI-64. Запропоноване рішення поєднує нейромережевий детектор YOLOv8, алгоритм DeepSORT, механізми апаратного прискорення та технологію INT8-квантування моделей.

Використання багатопотокової конвеєрної архітектури дозволило підвищити пропускну здатність системи та забезпечити стабільну обробку відеопотоку в режимі реального часу. Розроблений програмний модуль може бути використаний у системах інтелектуального відеоспостереження, транспортної аналітики, автономної робототехніки та інших прикладних рішеннях периферійного штучного інтелекту.

Список літератури

1. Redmon J., Farhadi A. YOLO: Real-Time Object Detection [Електронний ресурс]. URL: https://pjreddie.com/darknet/yolo/ (дата звернення: 18.05.2026)
2. Bewley A., Ge Z., Ott L., Ramos F., Upcroft B. Simple Online and Realtime Tracking. Proceedings of ICIP, 2016.
3. Wojke N., Bewley A., Paulus D. Simple Online and Realtime Tracking with a Deep Association Metric. Proceedings of ICIP, 2017.
4. Texas Instruments. Deep Learning (TIDL) User's Guide. 2023. 150 p. URL: https://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/linux/Foundational_Components/Machine_Learning/TIDL.html (дата звернення: 27.02.2026).
5. BeagleBone AI System Reference Manual. Texas Instruments [Електронний ресурс]. URL: https://github.com/beagleboard/beaglebone-ai/wiki/System-Reference-Manual (дата звернення: 27.02.2026).

Мамчур С. В.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Загородня Д.І., кафедра ІОСУ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ НА ОСНОВІ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПЛАТФОРМИ NVIDIA JETSON

Вступ. Сучасний розвиток систем комп'ютерного зору нерозривно пов'язаний із впровадженням технологій периферійних обчислень (Edge AI), які забезпечують виконання алгоритмів штучного інтелекту безпосередньо на вбудованих пристроях без передачі даних до хмарних сервісів. Такий підхід є особливо актуальним для систем відеоспостереження, автономної робототехніки, безпілотних апаратів та інтелектуальних IoT-пристроїв, де необхідно забезпечити мінімальні затримки обробки даних і високу швидкодію. Водночас використання сучасних згорткових нейронних мереж на вбудованих платформах супроводжується значними обчислювальними труднощами через обмежені ресурси процесора, пам'яті та енергоспоживання. Тому актуальним є розроблення програмних модулів класифікації зображень, адаптованих до архітектурних особливостей платформ NVIDIA Jetson [1–4].

Постанова задачі. Метою роботи є розробка програмного модуля класифікації зображень на основі згорткової нейронної мережі для вбудованої обчислювальної платформи NVIDIA Jetson із використанням методів оптимізації нейромережевого інференсу та апаратного прискорення. Об'єктом дослідження є процес класифікації зображень у системах комп'ютерного зору. Предметом дослідження є методи, алгоритми та програмні засоби оптимізації згорткових нейронних мереж для роботи на вбудованих обчислювальних платформах.

Основний матеріал. У рамках дослідження проведено аналіз сучасних архітектур згорткових нейронних мереж для систем Edge AI та особливостей їх використання на вбудованих платформах. Особливу увагу приділено платформі NVIDIA Jetson Nano, яка поєднує енергоефективність та можливість апаратного прискорення алгоритмів глибокого навчання [5].

Для забезпечення високої продуктивності розроблено архітектуру програмного модуля за конвеєрним принципом обробки даних. Архітектура включає підсистеми захоплення та попередньої обробки зображень, нейромережевої класифікації, постобробки результатів та взаємодії з користувачем. Загальну структуру розробленого програмного модуля наведено на рисунку 1.

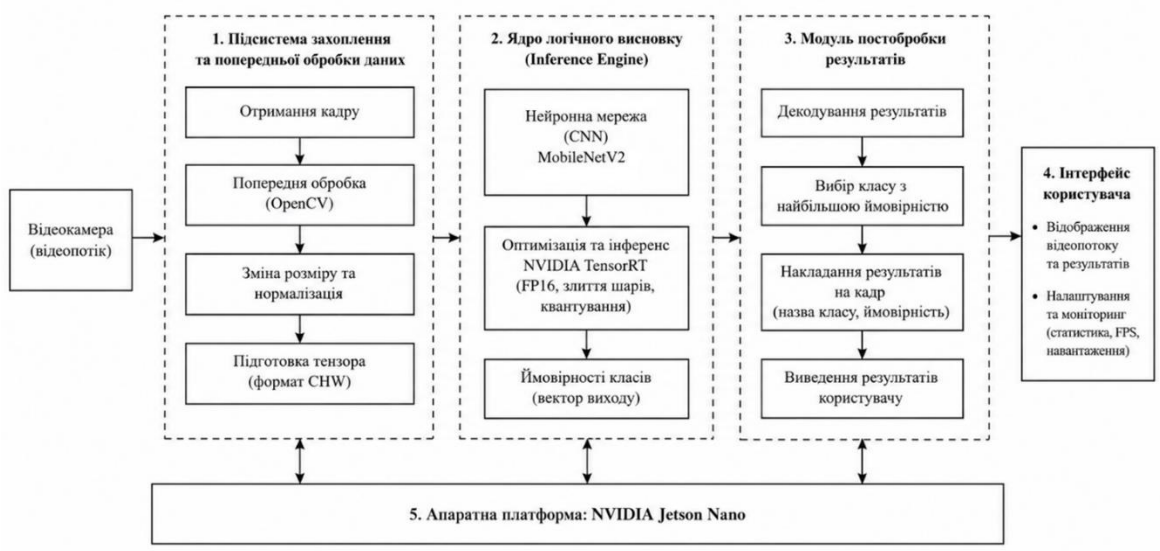


Рисунок 1 – Архітектурна схема програмного модуля класифікації зображень

Програмний модуль складається з підсистеми підготовки вхідних даних, ядра логічного висновку на основі згорткової нейронної мережі MobileNetV2 [3], оптимізованої за допомогою NVIDIA TensorRT [4], підсистеми постобробки результатів та інтерфейсу користувача. Основні обчислення виконуються безпосередньо на платформі NVIDIA Jetson Nano, що забезпечує локальну обробку даних без використання хмарних сервісів.

Для класифікації зображень використовується архітектура MobileNetV2, яка забезпечує ефективний баланс між точністю розпізнавання та обчислювальною складністю. Для підвищення швидкодії моделі застосовано механізми апаратної оптимізації TensorRT, які включають оптимізацію графа обчислень, злиття операцій та використання прискореного інференсу на графічному процесорі платформи NVIDIA Jetson.

Загальна структура програмного модуля включає такі функціональні блоки:

– блок захоплення зображень (забезпечує отримання вхідних даних від камери або відеопотоку);

– блок попередньої обробки (виконує нормалізацію, зміну розміру зображення та підготовку тензора для нейронної мережі);

– блок нейромережевої класифікації (реалізує логічний висновок на основі моделі MobileNetV2);

– блок постобробки результатів (виконує аналіз вихідних ймовірностей та формування прогнозованого класу);

– блок взаємодії з користувачем (забезпечує візуалізацію результатів класифікації).

Програмний модуль реалізовано мовою Python із використанням бібліотеки OpenCV, фреймворків TensorFlow/PyTorch та інструментарію NVIDIA TensorRT для прискорення виконання нейромережевого інференсу.

Проведені експериментальні дослідження підтвердили можливість виконання класифікації зображень у режимі реального часу на платформі NVIDIA Jetson Nano. Використання оптимізованої моделі та механізмів апаратного прискорення дозволило забезпечити високу швидкодію та достатню точність класифікації.

Висновки. У результаті проведеного дослідження виконано аналіз сучасних підходів до класифікації зображень у системах комп'ютерного зору та особливостей їх реалізації на вбудованих Edge AI-платформах. Розроблено архітектуру програмного модуля класифікації зображень для платформи NVIDIA Jetson, яка забезпечує ефективне використання апаратних ресурсів пристрою. Запропоноване рішення поєднує згорткову нейронну мережу MobileNetV2, механізми оптимізації TensorRT та конвеєрну архітектуру обробки даних.

Використання розробленого програмного модуля дозволяє виконувати класифікацію зображень у режимі реального часу без залучення хмарних сервісів. Результати роботи можуть бути використані у системах інтелектуального відеоспостереження, робототехніці, безпілотних системах та інших прикладних рішеннях периферійного штучного інтелекту.

Список літератури

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. URL: https://www.deeplearningbook.org/.

2. Lane N. D., Georgiev P. Can Deep Learning Revolutionize Mobile Sensing? Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications. 2015. P. 117–122.

3. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018. P. 4510–4520.

4. NVIDIA Corporation. NVIDIA TensorRT Documentation. URL: https://docs.nvidia.com/deeplearning/tensorrt/

5. NVIDIA Jetson Nano Developer Kit User Guide. URL: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

Цьома І.С.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Загородня Д.І., кафедра ІОСУ ЗУНУ

ВИЯВЛЕННЯ ДЕФЕКТІВ ЗВАРНИХ ШВІВ ІЗ ВИКОРИСТАННЯМ ГЛИБИННОГО НАВЧАННЯ

Вступ. Контроль якості зварних з'єднань є важливою складовою виробничих процесів у машинобудуванні, будівництві, енергетиці, транспортній та інших галузях промисловості. Надійність конструкцій значною мірою залежить від якості виконання зварних швів, оскільки дефекти зварювання можуть спричинити зниження міцності виробів, виникнення аварійних ситуацій та значні економічні збитки [1]. Традиційні методи неруйнівного контролю потребують залучення кваліфікованих фахівців і значних часових витрат, тому актуальним напрямом є автоматизація процесів контролю якості із застосуванням технологій комп'ютерного зору та глибинного навчання [2-4].

Постанова задачі. Основною метою роботи є розробка програмного модуля для автоматизованого виявлення дефектів зварних швів на основі методів глибинного навчання, що забезпечує підвищення точності та швидкості контролю якості зварних з'єднань за результатами аналізу цифрових зображень. Об'єктом дослідження є процес автоматизованого виявлення дефектів зварних швів на цифрових зображеннях. Предметом дослідження виступають методи комп'ютерного зору, алгоритми глибинного навчання та програмні засоби аналізу зображень, призначені для виявлення і класифікації дефектів зварних з'єднань.

Основний матеріал. У рамках дослідження проведено аналіз сучасних методів неруйнівного контролю та існуючих програмних систем автоматизованого виявлення дефектів зварних швів. Розглянуто сучасні програмні рішення автоматизованого виявлення дефектів зварних швів, що використовують технології комп'ютерного зору та штучного інтелекту. Дослідження охоплювало аналіз функціональних можливостей систем, методів обробки зображень, використання алгоритмів штучного інтелекту та особливостей інтеграції у виробничі процеси. За результатами аналізу визначено основні переваги та недоліки існуючих рішень, а також сформовано вимоги до розроблюваного програмного забезпечення.

Загальна структура програмного модуля включає клієнтський, серверний та нейромережевий рівні. Клієнтський рівень забезпечує взаємодію користувача із системою через кросплатформений клієнтський застосунок або вебінтерфейс. Серверний рівень відповідає за обробку запитів, підготовку даних та передачу результатів аналізу. Рівень машинного навчання виконує безпосереднє розпізнавання дефектів на зображеннях зварних швів з використанням нейронної мережі.

Головні функціональні блоки програмного модуля:

- Блок завантаження зображень. Забезпечує отримання зображень зварних швів для подальшого аналізу;
- Блок попередньої обробки даних. Виконує нормалізацію, масштабування та покращення якості вхідних зображень;
- Блок виявлення дефектів. Реалізує аналіз зображень за допомогою згорткової нейронної мережі архітектури YOLO;
- Блок візуалізації результатів. Забезпечує відображення знайдених дефектів та їх характеристик на вихідному зображенні.

Інформаційна взаємодія між компонентами системи здійснюється через REST API, що забезпечує передачу даних між клієнтською та серверною частинами.

Загальну структуру розробленого програмного модуля наведено на рисунку 1. Архітектура системи реалізована за клієнт-серверним принципом та включає клієнтську частину, серверний модуль обробки запитів, підсистему попередньої обробки зображень і модуль машинного навчання на базі нейромережі YOLO. Такий підхід забезпечує масштабованість системи, ізоляцію компонентів та можливість ефективного розгортання у контейнеризованому середовищі.

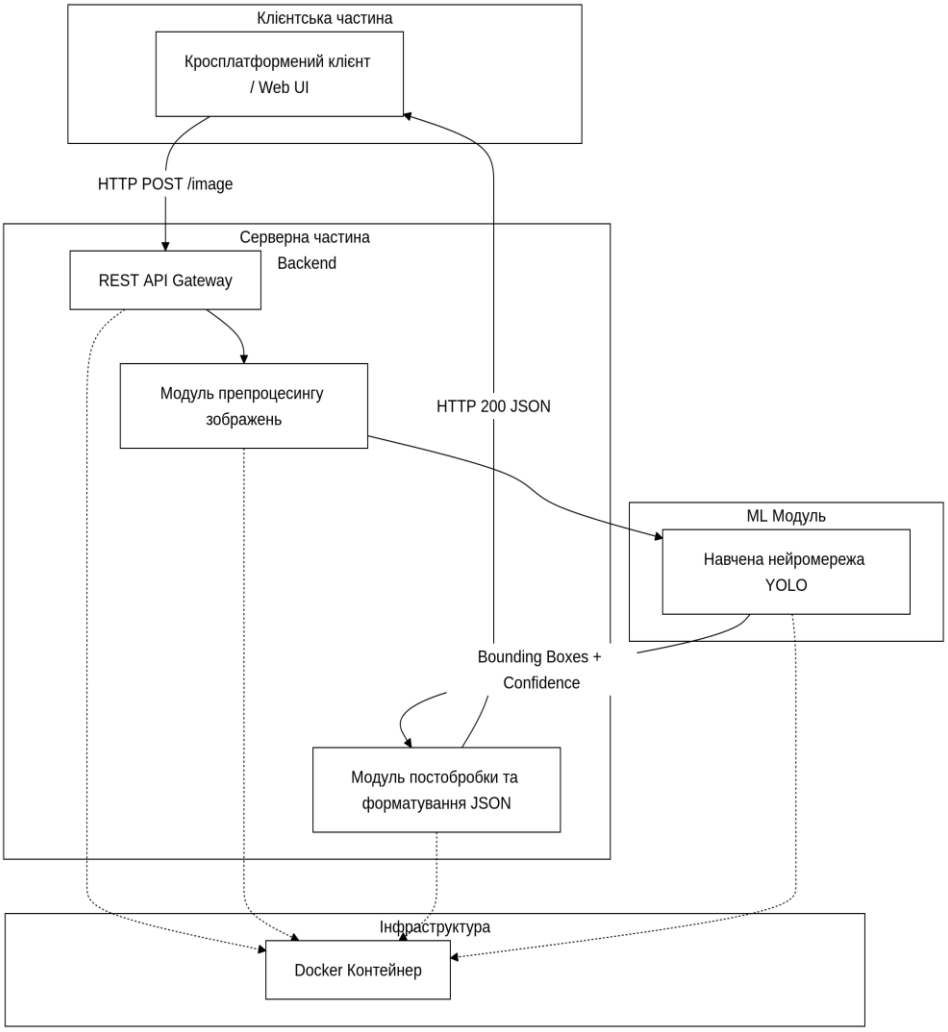


Рисунок 1 – Архітектура програмного модуля виявлення дефектів зварних швів на основі нейромережі YOLO

Для проєктування програмного забезпечення використано клієнт-серверний підхід із розділенням бізнес-логіки, інтерфейсу користувача та модуля машинного навчання. Основними сутностями системи є користувач, зображення, модель розпізнавання та результати детекції.

Розроблений програмний модуль дозволяє автоматично виявляти дефекти зварних швів на цифрових зображеннях, визначати їх тип та координати розташування. Для реалізації алгоритмів машинного навчання використано архітектуру YOLO, яка забезпечує високу швидкість та точність виявлення дефектів у режимі реального часу.

Серверна частина реалізована за допомогою фреймворку FastAPI, який забезпечує швидку обробку HTTP-запитів та інтеграцію з моделями штучного інтелекту. Для ізоляції середовища виконання та спрощення розгортання використано технологію Docker. Нейромережевий модуль розроблено мовою Python із застосуванням бібліотек PyTorch та Ultralytics [4, 5].

Для навчання моделі використано відкритий набір даних Welding Defect – Object Detection, що містить анотовані зображення зварних швів із різними типами дефектів. Попередня обробка зображень здійснювалася засобами бібліотеки OpenCV.

Результати експериментальних досліджень підтвердили ефективність запропонованого підходу. Середня точність виявлення дефектів (mAP@0.5) склала близько 88 %, а показник повноти для найбільш критичних дефектів досягнув 94 %. Час обробки одного зображення становить близько 300–350 мс, що дозволяє використовувати систему в умовах оперативного контролю якості.

Висновки. У результаті проведеного дослідження проаналізовано сучасні методи неруйнівного контролю та програмні рішення для автоматизованого виявлення дефектів зварних швів. На основі отриманих результатів сформовано вимоги до програмного модуля та спроєктовано його архітектуру. Реалізовано програмний модуль для автоматичного виявлення та класифікації дефектів зварних швів із використанням методів комп'ютерного зору та глибинного навчання.

Використання технологій Python, FastAPI, PyTorch, YOLO та Docker забезпечило високу продуктивність, масштабованість і зручність розгортання системи. Розроблений програмний модуль дозволяє автоматизувати процес контролю якості зварних з'єднань, мінімізувати вплив людського фактора та підвищити достовірність виявлення дефектів. Отримані результати підтверджують доцільність використання технологій глибинного навчання для вирішення задач промислової дефектоскопії та можуть бути використані як основа для подальшого розвитку інтелектуальних систем неруйнівного контролю.

Список літератури

6. ISO 6520-1:2023. Welding and allied processes – Classification of geometric imperfections in metallic materials. Geneva: ISO, 2023.
7. Bradski G., Kaehler A. Learning OpenCV 4 Computer Vision with Python. O'Reilly Media, 2021.
8. Redmon J., Farhadi A. YOLO: Real-Time Object Detection. URL: https://pjreddie.com/darknet/yolo
9. Ultralytics. YOLO Documentation. URL: https://docs.ultralytics.com
10. PyTorch Foundation. PyTorch Documentation. URL: https://pytorch.org/docs

---

Слотюк А.І.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Загородня Д.І., кафедра ІОСУ ЗУНУ

ПРОГРАМНИЙ МОДУЛЬ КЛАСИФІКАЦІЇ ПОШКОДЖЕНЬ БЕТОННИХ КОНСТРУКЦІЙ НА ОСНОВІ МЕТОДІВ ГЛИБИННОГО НАВЧАННЯ

Вступ. Забезпечення надійності та безпечної експлуатації будівельних споруд потребує регулярного контролю технічного стану бетонних і залізобетонних конструкцій. У процесі експлуатації під впливом механічних навантажень, атмосферних чинників, температурних коливань та корозійних процесів на поверхні конструкцій можуть виникати тріщини, відколи, відшарування бетону та інші дефекти. Традиційні методи діагностики базуються переважно на візуальному огляді експертами або використанні спеціалізованих засобів неруйнівного контролю, що потребує значних витрат часу та ресурсів. У зв'язку з розвитком технологій комп'ютерного зору та штучного інтелекту актуальним напрямом є створення автоматизованих систем аналізу цифрових зображень бетонних конструкцій, здатних виявляти та класифікувати дефекти їх поверхні.

Постановка задачі. Метою роботи є розробка програмного модуля автоматизованої класифікації пошкоджень бетонних конструкцій на основі методів глибинного навчання та мультимодальних моделей штучного інтелекту. Об'єктом дослідження є процес аналізу цифрових зображень бетонних поверхонь. Предметом дослідження виступають методи комп'ютерного зору, мультимодальні мовно-візуальні моделі та програмні засоби їх інтеграції у прикладні системи технічної діагностики.

Основний матеріал. У межах дослідження було проведено аналіз сучасних підходів до автоматизованого виявлення дефектів бетонних конструкцій. Розглянуто класичні згорткові нейронні мережі, методи Transfer Learning, алгоритми детекції об'єктів сімейства YOLO та сучасні мультимодальні моделі типу Vision-Language Model (VLM) [1-3].

Аналіз сучасних досліджень показав, що традиційні CNN-архітектури та моделі сімейства YOLO забезпечують ефективне виявлення дефектів бетонних конструкцій, однак не формують інтерпретованих текстових висновків для інженера-експерта. У зв'язку з цим перспективним є використання мультимодальних моделей класу Vision-Language Model, архітектура яких базується на поєднанні візуальних та мовних представлень даних [2].

Для реалізації програмного модуля обрано архітектурний підхід на основі моделей сімейств Qwen-VL та LLaVA [1, 3]. Використання відкритих моделей забезпечує можливість локального розгортання системи без передачі зображень до сторонніх хмарних сервісів, що є важливим для задач інженерної діагностики та моніторингу критичної інфраструктури.

На рисунку 1 наведено архітектуру запропонованого програмного модуля класифікації пошкоджень бетонних конструкцій на основі мультимодальних моделей класу Vision-Language Model. Система використовує підхід динамічного тайлінгу, за якого вхідне зображення бетонної конструкції розбивається на окремі фрагменти. Кожен тайл незалежно аналізується мультимодальною моделлю класу Vision-Language Model (VLM), після чого результати агрегуються та використовуються для формування структурованого висновку про технічний стан конструкції та наявність пошкоджень.

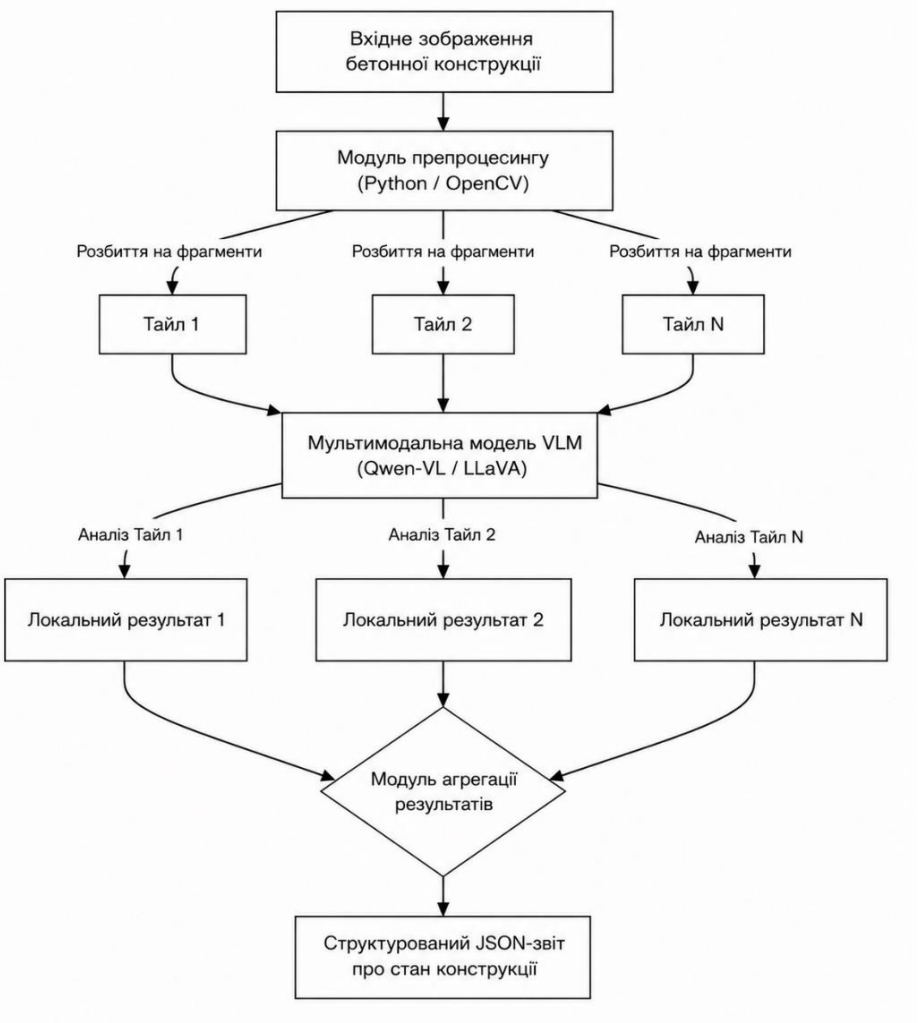


Рисунок 1 – Архітектура системи аналізу пошкоджень бетонних конструкцій на основі VLM

Однією з ключових проблем автоматизованого аналізу бетонних поверхонь є виявлення дрібних дефектів, зокрема мікротріщин, які можуть втрачатися під час масштабування зображень для обробки нейронними мережами. Для вирішення цієї проблеми запропоновано використання механізму динамічного тайлінгу, за якого вхідне зображення розбивається на набір фрагментів, що незалежно аналізуються мультимодальною моделлю Qwen-VL або

LLaVA. Отримані локальні результати агрегуються для формування підсумкового висновку про технічний стан конструкції та структурованого JSON-звіту.

Програмний модуль реалізовано мовою Python із використанням фреймворку FastAPI. Серверна частина забезпечує прийом зображень через REST API, виконання інференсу мультимодальної моделі та формування структурованого JSON-звіту. Для зручності взаємодії з користувачем реалізовано веб-інтерфейс на основі бібліотеки Gradio, який дозволяє завантажувати фотографії конструкцій та переглядати результати аналізу у браузері.

Для забезпечення переносимості та спрощення розгортання програмного забезпечення використано технологію контейнеризації Docker. Конфігурація системи дозволяє запускати модуль як на локальних робочих станціях інженерів, так і на серверному обладнанні у складі систем моніторингу технічного стану споруд.

Для оцінювання запропонованого підходу використано відкриті набори даних SDNET2018 [4], CODEBRIM [5] та Concrete Crack Images for Classification [6].

Проведено функціональне тестування програмного модуля, яке підтвердило коректність роботи механізмів обробки зображень, мультимодального аналізу та формування структурованого JSON-звіту. Проведений аналіз показав перспективність використання VLM-моделей для автоматизованої інженерної діагностики та підтримки прийняття рішень під час обстеження будівельних об’єктів.

Висновки. У результаті проведеного дослідження виконано аналіз сучасних методів автоматизованого виявлення та класифікації пошкоджень бетонних конструкцій. Обґрунтовано доцільність використання мультимодальних моделей штучного інтелекту для задач інженерної діагностики. Запропоновано підхід до класифікації пошкоджень бетонних конструкцій на основі мультимодальних моделей класу VLM із використанням механізму динамічного тайлінгу зображень. Такий підхід забезпечує аналіз зображень та формування структурованих результатів для підтримки прийняття рішень під час технічного обстеження будівельних конструкцій.

Список літератури

1. Liu H., Li C., Wu Q., Lee Y. J. Visual Instruction Tuning // Advances in Neural Information Processing Systems (NeurIPS). 2023. Retrieved June 14, 2026, from https://arxiv.org/abs/2304.08485
2. Radford A., Kim J. W., Hallacy C. et al. Learning Transferable Visual Models From Natural Language Supervision // Proceedings of the 38th International Conference on Machine Learning (ICML). 2021. Retrieved June 14, 2026, from https://arxiv.org/abs/2103.00020
3. Bai J., Bai S., Yang S. et al. Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities. 2023. Retrieved June 14, 2026, from https://arxiv.org/abs/2308.12966
4. Dorafshan S., Thomas R. J., Maguire M. SDNET2018: An Annotated Image Dataset for Non-Contact Concrete Crack Detection Using Deep Convolutional Neural Networks // Data in Brief. 2018. Retrieved June 14, 2026, from https://doi.org/10.1016/j.dib.2018.04.096
5. Mundt M., Majumder S., Murali S., Panetsos P., Ramesh V. Meta-Learning Convolutional Neural Architectures for Multi-Target Concrete Defect Classification with the COncrete DEfect BRidge IMage Dataset // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, 2019. P. 11196–11205.
6. Özgenel Ç. F. Concrete Crack Images for Classification. Mendeley Data. 2019. V. 2. DOI: 10.17632/5y9wdsg2zt.2. URL: https://doi.org/10.17632/5y9wdsg2zt.2

Божко М.В.
Студент 4 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н., доцент Биковий П.Є., кафедра ІОСУ ЗУНУ

МОБІЛЬНИЙ ЗАСТОСУНОК З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ ВІЗУАЛІЗАЦІЇ МУЗЕЙНОГО КОНТЕНТУ З УРАХУВАННЯМ ПОТРЕБ ОСІБ З ПОРУШЕННЯМИ СЛУХУ

Вступ. Сучасні музеї активно впроваджують цифрові технології для підвищення рівня взаємодії відвідувачів з експозиціями та забезпечення доступності культурної спадщини для різних категорій користувачів. Особливо актуальною є проблема надання музейної інформації особам з порушеннями слуху, для яких традиційні аудіогіди та усні пояснення екскурсоводів є малодоступними. Одним із перспективних напрямів вирішення цієї проблеми є використання технологій доповненої реальності (Augmented Reality, AR), які дозволяють інтегрувати цифровий контент безпосередньо у фізичний простір музею та забезпечувати його візуальне представлення у зручній для користувача формі [1-3].

Постановка задачі. Метою роботи є розробка мобільного застосунку на основі технологій доповненої реальності для візуалізації музейного контенту з урахуванням потреб осіб з порушеннями слуху. Об’єктом дослідження є процеси надання та візуального сприйняття цифрового контенту в інклюзивному музейному середовищі. Предметом дослідження виступають методи, алгоритми та програмні засоби доповненої реальності, комп’ютерного зору та інтелектуальної генерації контенту для мобільних застосунків.

Основний матеріал. У межах дослідження проведено аналіз сучасних технологій цифрової трансформації музейного простору та інклюзивних підходів до представлення культурного контенту. Розглянуто сучасні програмні рішення для цифрової трансформації музейного простору, що використовують технології комп’ютерного зору, доповненої реальності та мультимедійної візуалізації для супроводу музейних експозицій [1–3].

На основі проведеного аналізу встановлено, що більшість існуючих систем орієнтовані на загальну аудиторію та не забезпечують повною мірою адаптацію контенту до потреб осіб з порушеннями слуху. У зв’язку з цим запропоновано архітектуру мобільного застосунку, що поєднує технології доповненої реальності, комп’ютерного зору та інтелектуальної генерації текстового контенту.

На рисунку 1 наведено архітектуру розробленого мобільного застосунку для візуалізації музейного контенту з використанням технологій доповненої реальності. Система складається з мобільного клієнта, який містить модулі доповненої реальності, розпізнавання експонатів, доступності та управління контентом, а також серверної підсистеми на базі FastAPI, що включає RAG-модуль, сервіс генерації контенту, сервіс розпізнавання зображень та підсистему потокової передачі даних через WebSocket.

Архітектура системи включає клієнтську частину, реалізовану на базі Flutter, серверну підсистему обробки даних та модуль доповненої реальності [5]. Розпізнавання музейних експонатів виконується за допомогою алгоритмів комп’ютерного зору та моделей машинного навчання, після чого система отримує відповідний опис експоната та формує адаптований інформаційний контент.

Однією з ключових особливостей розробленого рішення є використання просторової візуалізації текстової інформації безпосередньо біля експоната у середовищі доповненої реальності [3]. Такий підхід дозволяє користувачу одночасно спостерігати музейний об’єкт і отримувати текстові пояснення без необхідності переключення уваги між різними джерелами інформації.

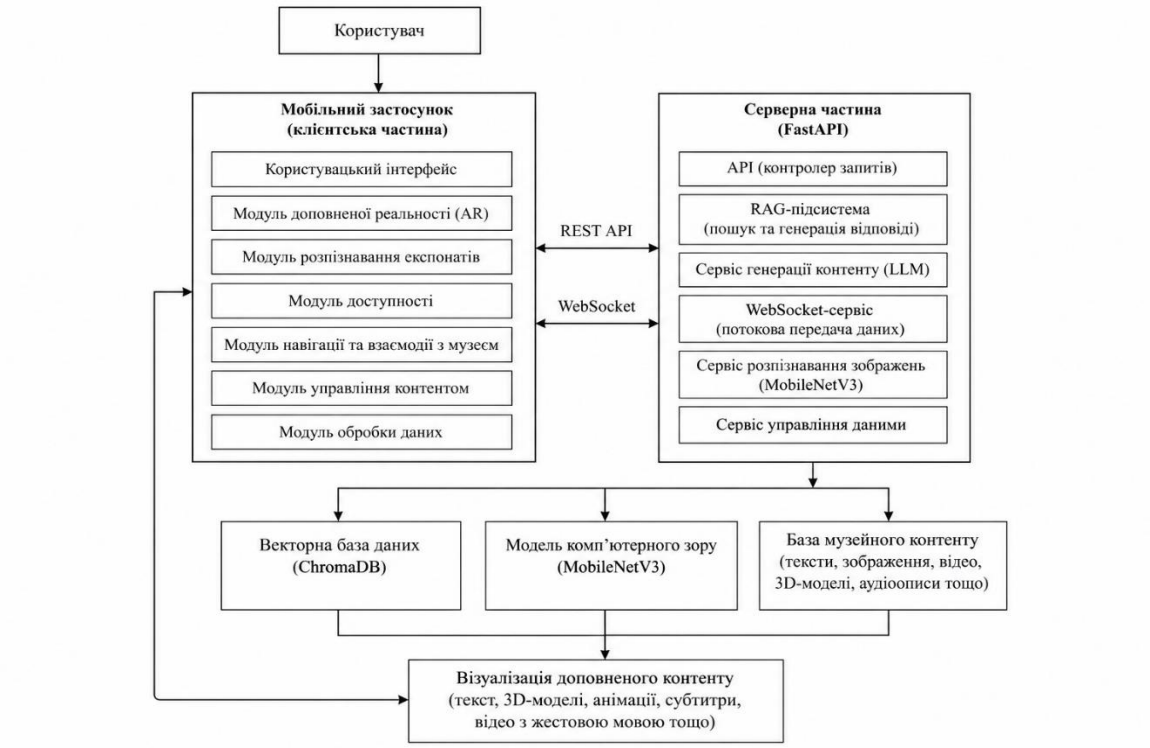


Рисунок 1 – Архітектура мобільного застосунку для візуалізації музейного контенту з використанням технологій доповненої реальності

Для забезпечення масштабованості системи реалізовано серверну архітектуру на базі сучасних вебтехнологій із підтримкою потокової передачі даних та інтеграцією механізмів Retrieval-Augmented Generation (RAG) [4]. Це дозволяє формувати контекстно-залежні пояснення та адаптувати зміст повідомлень до потреб користувача.

Проведено функціональне тестування мобільного застосунку, AR-модуля та серверної підсистеми. Результати експериментального дослідження підтвердили коректність роботи механізмів розпізнавання експонатів, відображення AR-контенту та генерації адаптованих текстових повідомлень в умовах реального музейного середовища.

Висновки. У результаті дослідження виконано аналіз сучасних підходів до використання технологій доповненої реальності в музейній сфері та обґрунтовано доцільність їх застосування для створення інклюзивного середовища для осіб з порушеннями слуху. Розроблено архітектуру мобільного застосунку, що поєднує засоби комп'ютерного зору, доповненої реальності та інтелектуальної генерації контенту. Запропоноване рішення сприяє підвищенню доступності музейного контенту та покращенню взаємодії осіб з порушеннями слуху з музейними експозиціями.

Список літератури

1. Mannion S. Smartify: Connecting Audiences with Art Through Image Recognition Technology. URL: https://smartify.org

2. Azuma R. T. A Survey of Augmented Reality // Presence: Teleoperators and Virtual Environments. 1997. Vol. 6, No. 4. P. 355–385.

3. Billinghurst M., Clark A., Lee G. A Survey of Augmented Reality // Foundations and Trends in Human–Computer Interaction. 2015. Vol. 8, No. 2–3. P. 73–272.

4. Lewis P., Perez E., Piktus A. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems (NeurIPS). 2020. URL: https://arxiv.org/abs/2005.11401

5. Google. Flutter – Build apps for any screen. URL: https://flutter.dev