Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Аналіз руйнувань міської інфраструктури
та визначення черговості її відновлення
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШІ-21-1
Наталія Пиріг
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Штучний інтелект
(повна назва освітньої програми)
Керівник доц. Лариса Чала
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Пиріг Наталії Янівні
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз руйнувань міської інфраструктури та визначення черговості її відновлення

затверджена наказом університету від 19 травня 20 25 р. № 378Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17 червня 20 25 р.

3. Вихідні дані до роботи Офіційна документація та посібники для програмування мовою Python та використання його її бібліотек, документація фреймворку Flask, офіційна документація для YOLO та Detectron2, відомості про типові проєкти житлових та громадських будівель України, архітектурні бази даних та переліки об'єктів культурної спадщини, загальнодоступні набори зображень з наслідками руйнувань інфраструктури, наукові публікації та веб-ресурси.

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та постановка задачі
2) Теоретичні дослідження
3) Проектування системи та її розробка
4) Експериментальні дослідження

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	19.05.2025	виконано
2	Аналіз предметної галузі	21.05.2025	виконано
3	Огляд існуючих систем та застосунків	22.05.2025	виконано
4	Концептуальна розробка системи	22.05.2025	виконано
5	Визначення вимог до розроблювальної системи	22.05.2025	виконано
6	Побудова та анотування набору зображень	24.05.2025	виконано
7	Дослідження особливостей сегментації	25.05.2025	виконано
	екземплярів		
8	Дослідження шляхів боротьби з дисбалансом	25.05.2025	виконано
	класів		
9	Вибір моделей сегментації екземплярів та	27.05.2025	виконано
	проведення їхнього порівняльного аналізу		
10	Ознайомлення з підходами до ранжування	27.05.2025	виконано
11	Пошук та отримання необхідної інформації про	27.05.2025	виконано
	будівлі Харкова		
12	Практична реалізації системи	30.05.2025	виконано
13	Тестування системи	31.05.2025	виконано
14	Написання пояснювальної записки	01.05.2025	виконано
15	Попередній захист	07.05.2025	виконано
16	Захист перед екзаменаційною комісією	17.06.2025	виконано

Дата видачі завдання 19 травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Лариса Чала
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 121 с., 28 рис., 1 табл., 4 дод., 56 джерел.

ВИЗНАЧЕННЯ ПРІОРИТЕТІВ, ВІДНОВЛЕННЯ БУДІВЕЛЬ, ІНФРАСТРУКТУРА, КОМП'ЮТЕРНИЙ ЗІР, КУЛЬТУРНА СПАДЩИНА, МЕТОДИ БАГАТОКРИТЕРІАЛЬНОГО ПРИЙНЯТТЯ РІШЕНЬ, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ.

Об'єкт дослідження – процес аналізу руйнувань міської інфраструктури та визначення пріоритетності її відновлення під час воєнного стану та після його завершення.

Предмет дослідження – методи автоматизованого аналізу пошкоджень будівель за допомогою комп'ютерного зору та алгоритмів ранжування для визначення пріоритетності відновлення.

Мета роботи – розробити концепцію та реалізувати прототип системи, що дозволить формувати список будівель для першочерговості відновлення з урахуванням ступеню зруйнованості споруди на основі автоматичного виявлення та класифікації пошкоджень, соціальної та культурної значущості.

Методи дослідження – порівняльний аналіз прикладних рішень, що реалізують функціонал виявлення пошкоджень міської інфраструктури та підтримки прийняття рішень щодо її відновлення.

У роботі розглядається актуальна проблема автоматизованого аналізу пошкоджень міської інфраструктури в умовах воєнного конфлікту та визначення пріоритетності її відновлення. Запропоновано концепцію системи для автоматичного виявлення пошкоджень будівель, оцінки ступеню пошкодження будівель та формувати списки пріоритетів розглянутих будівель. Розроблено прототип зазначеної системи.

ABSTRACT

Bachelor’s thesis contains: 121 pp., 28 fig., 1 tabl., 4 ann., 56 references.

BUILDING RESTORATION, COMPUTER VISION, CULTURAL HERITAGE, DETERMINATION OF PRIORITIES, IMAGE SEGMENTATION, INFRASTRUCTURE, MULTI-CRITERIA DECISION MAKING METHODS.

Object of study – the process of analyzing damage to urban infrastructure and determining priorities for its restoration during martial law and after its termination.

Subject of study – methods of automated analysis of building damage using computer vision and ranking algorithms to determine restoration priorities.

Aim of the work – to develop a concept and implement a prototype of a system that allows the formation of lists of buildings prioritized for restoration, taking into account the degree of structural damage based on automatic detection and classification of damages, as well as social and cultural significance.

Research methods – comparative analysis of applied solutions that implement functionality for detecting damage to urban infrastructure and supporting decision-making regarding its restoration.

This work addresses the urgent problem of automated analysis of urban infrastructure damage in the context of armed conflict and the determination of priorities for its restoration. A concept of a system for automatic detection of building damages, assessment of damage severity, and formation of priority lists of considered buildings is proposed. A prototype of the mentioned system has been developed.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 8
Вступ 9
1 Аналіз предметної галузі та постановка задачі 11
1.1 Опис предметної галузі 11
1.2 Опис проблем та викликів 13
1.3 Огляд існуючих застосунків 16
1.4 Інноваційність та переваги розроблюваної системи 26
1.5 Постановка задачі 29
2 Теоретичні дослідження 31
2.1 Огляд існуючих досліджень 31
2.2 Огляд та порівняння напрямків комп’ютерного зору 41
2.3 Порівняння моделей сегментації екземплярів 46
2.4 Огляд моделей YOLOv11 та Mask-R-CNN 48
2.5 Шляхи вирішення проблеми дисбалансу класів 51
2.6 Методи ранжування 56
3 Проектування системи та її розробка 64
3.1 Проєктування системи 64
3.2 Структура проєкту 67
3.3 Підготовка набору даних 70
3.4 Модуль визначення пошкоджень 76
3.5 Модуль розрахунку оцінок для будівлі 78
3.6 Модуль ранжування об'єктів 80
3.8 Серверна частина та API 83
3.9 Інтерфейс користувача 86
4 Експериментальні дослідження 91
4.1 Аналіз моделей і методів боротьби з дисбалансом класів 91
4.2 Перспективи розвитку 99
Висновки 102

Перелік джерел посилання 103
Додаток А Код функцій розрахунку оцінок……..110
Додаток Б Код маршрутів серверної частини Flask……..116
Додаток В Код кастомного DataLoader для Mask R-CNN……120
Додаток Г Відомість кваліфікаційної роботи 121

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AHP – Analytic Hierarchy Process – аналітичний ієрархічний процес;

ANN – Artificial Neural Network – штучна нейронна мережа;

BWM – Best Worst Method – метод «найкращий–найгірший»;

CAM – Class Activation Mapping – карта активації класів;

CNN – Convolutional Neural Network – свёрточна нейронна мережа;

DDM – Damage Detection Map – карта виявлення пошкоджень;

FPN – Feature Pyramid Network – мережа пірамідальних ознак;

IoU – Intersection over Union – коефіцієнт перетину над об’єднанням;

LTR – Learning to Rank – навчання ранжуванню;

MCDA – Multi-Criteria Decision Analysis – багатокритеріальний аналіз рішень;

MCDM – Multi-Criteria Decision Making – багатокритеріальне прийняття рішень;

NDCG – Normalized Discounted Cumulative Gain – нормалізована дисконтована кумулятивна виграш;

YOLO – You Only Look Once – алгоритм одноразового виявлення об’єктів.

ВСТУП

Унаслідок повномасштабного вторгнення, розпочатого у 2022 році, інфраструктура міст України зазнала значних руйнувань по всій її території, що призвело до порушення нормального функціонування міст, погіршення якості життя мешканців та втрати культурно-історичної спадщини [1]. Через це гостро постала потреба у швидкому та ефективному відновленні пошкоджених об’єктів міського середовища. Однак у сучасних умовах обмежених ресурсів неможливо відновити всі зруйновані об’єкти одночасно, тому особливого значення набуває проблема визначення пріоритетності відновлення.

Визначення черговості відновлення потребує врахування різних критеріїв: ступеня фізичних пошкоджень, конструктивних особливостей будівель, їхньої соціальної ролі та культурно-історичної цінності. При цьому важливу роль відіграє прозорість процесу прийняття об’єктивних рішень. Традиційні методи оцінки руйнувань, що базуються на експертних оглядах, є трудомісткими, повільними та часто суб’єктивними, особливо в умовах великої кількості пошкоджених об’єктів на території всієї країни. Гостро постає потреба в зміні підходів до оцінки ступеню пошкодження інфраструктури та планування проведення відновлювальних робіт.

На даний момент в Україні відсутні такі інформаційні системи, що здатні автоматизувати процеси отримання всієї необхідної для прийняття рішення інформації про будівлю, оцінити серйозність руйнування та запропонувати комплексний підхід для визначення пріоритетності відновлення споруд з урахуванням різних факторів, враховуючи не лише фізичний стан об’єктів, але й їх соціальну важливість та історико-культурну цінність. Більшість існуючих рішень обмежується оцінкою лише зовнішніх ознак ушкоджень і не здатні надавати рекомендації щодо черговості відновлення на основі багатофакторного аналізу.

У зв’язку з зазначеним особливо актуальним є створення інтелектуальної системи, яка на основі даних архітектурно-технічної документації, метаданих про будівлі та з використанням методів комп’ютерного зору здатна визначати пріоритети відновлення об’єктів міської інфраструктури. Розробка та впровадження подібної системи дозволить не тільки підвищити ефективність використання наявних ресурсів, але й забезпечить соціально справедливу стратегію відновлення, яка буде враховувати потреби громади, важливість функцій об’єкта та збереження національної пам’яті.

У рамках даної роботи досліджується предметна область, пов’язана з аналізом руйнувань міської інфраструктури, визначаються основні проблеми та виклики, що виникають під час оцінки пошкоджень і планування відновлення. Особлива увага приділяється інноваційним аспектам розроблюваної системи, яка базується на використанні методів та моделей комп’ютерного зору для автоматичного виявлення пошкоджень будівель за даними з зображень.

Отже, дана робота має значний потенціал для практичного застосування. Розроблена в рамках цієї роботи система може бути використана місцевими адміністраціями, міжнародними організаціями чи фондами для планування відновлення міст, що зазнали значних руйнувань. Автоматизація процесу аналізу та ранжування об’єктів сприятиме раціональному розподілу ресурсів та прискоренню відбудови. Крім того, система сприятиме збереженню культурної спадщини, що є важливим для підтримки національної ідентичності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Від початку повномасштабного вторгнення у 2022 року в Україні було повністю зруйновано або частково пошкоджено велику кількість, як житлових, так і не житлових будівель. Згідно зі звітом, наданим Київською школою економіки [2], з лютого 2022 року і по січень 2024 року було зруйновано або пошкоджено близько 250 тис. об'єктів житлового фонду, 1284 закладів охорони здоров'я, 3,8 тис. освітніх закладів, 160 об'єктів соціального захисту населення, 426 великих та середніх будівель приватних підприємств, 2,9 тис. торгових точок до яких відносяться магазини, аптеки, кіоски, автозаправні станції. За наданими у звіті даними Харківська область входить до трійки найбільш пошкоджених областей України.

Крім значної фінансової та соціальної шкоди, країні було завдано значної культурної шкоди. За оцінкою комітету Всесвітньої Спадщини ЮНЕСКО [3] станом на 16 квітня 2025 року в Україні було зафіксовано та підтверджено пошкодження 485 об'єктів спадщини, серед яких 257 будівлям, які становлять історичний та/або мистецький інтерес, 149 релігійним об'єктам, 88 музеям, пам'ятникам, бібліотекам, архівам та археологічним об'єктам. При цьому найбільш пошкодженими областями є Донецька (132 пам'ятки) та Харківська (81 пам'ятка) області.

Перед державою постає пріоритетною задача відновлення зруйнованих і пошкоджених споруд як під час війни, так і в постконфліктний період. З огляду на величезні обсяги зруйнованої інфраструктури, а також обмежені фінансові та робочі ресурси і продовження воєнних дій, що становлять не лише загрозу для робітників, які виконують реставраційні роботи, але й появу нових руйнувань, стає очевидною неможливість швидкого відновлення всіх пошкоджених об'єктів та виникнення завдання визначення пріоритетів відновлення споруд.

На сьогоднішній день в країні розробляються проєкти та плани для відновлення пошкоджених споруд, також проводяться відновлювальні роботи. Процес реконструкції споруди, як правило, складається з декількох етапів, таких як оцінювання ушкоджень та визначення її ремонтопридатності, розробка проєкту реконструкції, затвердження розробленого проєкту відновлення та отримання всіх необхідних дозволів, проведення будівельних робіт, прийняття відреставрованого об'єкта та введення його в експлуатацію [4]. Даний процес займає значний час, а також потребує участі спеціалізованих фахівців, яких може бути обмежена кількість, значно затримується через бюрократичну тяганину.

Важливим недоліком процесу відновлення є, як правило, відсутності повних і чітких критеріїв визначення пріоритетності відновлення будівель з огляду на різноманітні чинники, які можуть вплинути на цінність об’єкту. Під час відбудови житлових та нежитлових споруд, що не належать до критичної інфраструктури, яка завжди підлягає відновленню в першу чергу, виникає необхідність у підтримці балансу між культурно-історичною цінністю будівлі, а також її функціональним значенням і впливом на життя суспільства. Також суттєвим недоліком є виключення громади з процесу обговорення пріоритетності проведення відновлювальних робіт або ненадання інформації про причини вибору тієї чи іншої споруди для першочерговості реконструкції за умови обмеженого фінансування. Наприклад, місцева влада Харкова оголосила про плани на відновлення близько 200 пошкоджених будівель до 2025 року [5], однак при цьому не було надано коментарів щодо причин вибору саме цих споруд. Такий підхід може викликати у місцевих мешканців питання про об'єктивність вибору об'єктів для відновлення.

Таким чином процес вибору споруди для проведення відновлювальних робіт, перетворюється на складне завдання, вирішення якого вимагає врахування як рівня фізичних пошкоджень, так і соціального та культурного значення будівель. Отже, виникає потреба у створенні

інтелектуальної системи, що дозволила б автоматизувати процеси визначення пошкоджених споруд, оцінювання серйозності їхньої ушкодженості, та визначення пріоритетів відновлення з урахуванням різних факторів. Такі системи дали б змогу зробити процес вибору об'єктів для першочергового відновлення більш об'єктивним і прозорим.

1.2 Опис проблем та викликів

При розробці систем для виявлення пошкоджень будівель у контексті воєнних конфліктів, а також для визначення пріоритетів відновлення з урахуванням ступеня пошкоджень та культурно-історичної цінності об’єктів, розробники стикаються з низкою проблем та викликів.

Однією із основних проблем, що постає, є брак достатньої кількості якісних зображень, на яких можна навчити моделі для автоматизованого виявлення та оцінки пошкоджень. На сьогодні є декілька великих наборів даних, що відображають пошкодження будівель в наслідок природних лих, наприклад xBD [6]. Зазначений набір даних містить понад 850000 анотацій будівель на супутникових знімках до і після природних катастроф таких, як урагани і землетруси, з мітками рівня пошкоджень. На рисунку 1.1 представлено декілька пар зображень з зазначеного набору даних.

Моделі, навчені на такому наборі даних, при використані на практиці вимагають наявності супутникових знімків до і після появи руйнувань для аналізу пошкоджених будівель, що пов'язано з високими витратами на отримання таких даних та залежністю від зовнішнього обладнання, такого як супутники. В деяких випадках використання такої моделі стає неможливим через відсутність знімків до руйнувань.

Точність побудованих на xBD моделей при застосуванні до воєнних ушкоджень може бути знижена, оскільки супутникові зображення часто не фіксують дрібні дефекти, такі як тріщини, а також через відмінності в характері ушкоджень: воєнні руйнування, наприклад, ушкодження фасадів

від обстрілів, відрізняються від наслідків стихійних лих, і не відображені в відповідних наборах даних. Таким чином не всі типи пошкоджень можуть бути виявлені побудованими моделями.

Рисунок 1.1 – Вигляд місцевості до (верхній ряд) і після (нижній ряд) катастрофи з набору даних xBD

Крім того, вплив довкілля, значущий під час аналізу затоплень або пожеж, може спотворювати результати, оскільки моделі, навчені на даних про природні катастрофи, орієнтуються на зміни ландшафту, які в умовах воєнних дій можуть бути відсутніми або не корелювати з ушкодженнями будівель [7]. Серйозну проблему також становить точковий характер руйнувань отриманих в ході воєнних дій, який вимагає детальнішого оцінювання окремих об'єктів, на відміну від узагальненого підходу до зон, що постраждали від стихійних лих.

Загалом точність оцінки пошкоджень значною мірою залежить від якості даних, що подаються на вхід. Будівлі знаходяться поруч з іншими будинками, а також мають оточуюче середовище, яке може заважати моделі правильно визначити ключові ознаки на зображеннях. При аналізі

супутникових знімків можна зустрітись з проблемою наявності хмар, що частково чи повністю загороджують будинки.

Інша проблема полягає в неможливості самостійно набрати набір даних достатніх розмірів для навчання моделі. Продовження ведення воєнних дій ставить під загрозу життя фахівців, що повинні на місті оцінювати у високій якості зображень, як для навчання моделі, так і для її роботи, що може значно впливати на точність передбачень моделі. Тіні, дерева, погодні умови або дим можуть спотворювати дані, що особливо проблематично при використанні зйомок з дронів [8]. Також варто враховувати, що пошкодження будівель з різних ракурсів можуть виглядати по-різному. Одне й те саме пошкодження, сфотографоване з різного ракурсу, може бути по-різному розпізнане моделлю.

Важливою проблемою для побудови необхідної системи є необхідність урахування того факту, що будинки будуються з різних матеріалів (наприклад, бетон, цегла, моноблок), що впливає на їхню стійкість і доцільність відновлення [9]. Наприклад, бетонні конструкції можуть витримувати певні навантаження, тоді як цегляні стіни більш уразливі до вибухових хвиль. Також тріщини в цегляній будівлі можуть бути не настільки небезпечними для поводження, як у будівлях виготовлених з бетону. Ці відмінності ускладнюють розробку універсальних моделей оцінки збитку. При виявленні пошкоджень та оцінці ступеня руйнування будівель критично важливо враховувати, які стіни є несучими, оскільки сильне пошкодження ненесучої стіни може становити меншу загрозу для стійкості конструкції, ніж незначне пошкодження несучої стіни, яке здатне порушити цілісність усієї будівлі. Також необхідно брати до уваги стан перекриттів, оскільки їхні ушкодження можуть істотно впливати на безпеку споруди, що робить недостатньою тільки зовнішню оцінку або візуальний огляд. Для точної оцінки серйозності пошкоджень часто потрібне застосування спеціалізованого обладнання, такого як ультразвукові сканери, лазерні далекоміри або георадари, які дають змогу

виявити приховані дефекти в конструкціях і визначити їхній вплив на загальну стабільність будівлі [10].

Країни, які постраждали від військових конфліктів, часто стикаються з дефіцитом фінансових і людських ресурсів. В Україні, наприклад, відновлення інфраструктури вимагає значних інвестицій, які обмежені через триваючий конфлікт [11]. Це ускладнює впровадження сучасних дорогих технологій, таких як ШІ або дрони, для оцінювання збитків.

Важливою проблемою є відсутність системи оцінки культурного значення. Визначення пріоритетів відновлення ускладнюється відсутністю об'єктивних критеріїв для оцінки культурної або історичної значущості будівель. Наприклад, немає стандартизованих методів для порівняння цінності житлових будинків, шкіл або пам'яток архітектури, що ускладнює ухвалення рішень щодо того, які об'єкти відновлювати в першу чергу [12].

1.3 Огляд існуючих застосунків

Після початку війни в Україні в 2014 року, особливо після повномаштабного вторгнення у 2022 року в країні з’явилась велика кількість пошкоджених будівель. Слід зазначити, що ці будинки знаходяться в різних куточках країни, з’являються в різні часові періоди та мають різний рівень серйозності пошкоджень. З’явилась нагальна потреба в автоматизованому способу локалізації пошкоджених будинків та оцінка серйозності їх пошкоджень з подальшою метою планування їхнього відновлення. На сьогодні існує ряд ініціатив та платформ, що вже використовуються або можуть бути використані в Україні для виконання описаного завдання. Далі наводиться огляд таких інструментів.

На початку повномасштабної війни в 2022 році ООН в рамках програми UNDP для України запустила проект «Infrastructure Semantic Damage Detector» [13] в рамках якого було створено модель, що використовує алгоритми машинного навчання та методи обробки великих

даних для аналізу пошкоджень інфраструктури шляхом обробки звітів. Побудована модель в якості джерела даних використовує ACLED, що є відкритою базою даних із глобальними реальними даними про конфлікти. Система створювалася для визначення місця розташування пошкоджень інфраструктури та масштабу завданих збитків для швидкого реагування. Також передбачалося використання цієї системи для надання готових звітів для формування запитів про допомогу і розрахунку її розміру в низку міжнародних гуманітарних організацій. Система не є комерційною, у відкритому доступі її знайти не вдалось.

Шляхом аналізу текстів звітів добуваються дані про дату, час, місце, причину та тип пошкодження. Однією з проблем, яку вирішили розробники системи – це визначення типу ушкодженої інфраструктури, враховуючи, що в звітах не завжди вказувався її тип в потрібному форматі. Було запропоновано дев’ять категорій, за якими йшла класифікація: промислові об’єкти, логістика, енергетика/електропостачання, телекомунікації, сільське господарство, охорона здоров’я, освіта, житло та бізнес. Шляхом визначення для кожної з зазначених категорій ключових слів та використання косинусної міри схожості для встановлення семантичного зв’язку між текстом звітів і категоріями інфраструктури побудована система може автоматично класифікувати типу пошкодженої інфраструктури без явного вказання про це в інформаційних записах.

Фінальним етапом роботи системи є візуалізація отриманих результатів на інтерактивній карті України. Можливість фільтрувати виявлені та відображені та карті будинки за областями та населеними пунктами їх находження, а також за типом їх функціонування, дозволяє ознайомитись з географічним розподіл отриманої шкоди інфраструктури різного типу. Можливість побудови стовпчастих та пелюсткових діаграми дозволяє користувачам більш детально ознайомитися з отриманими результатами. На рисунку 1.2 наведено вигляд побудованого інтерактивної дошки з отриманими результатами.

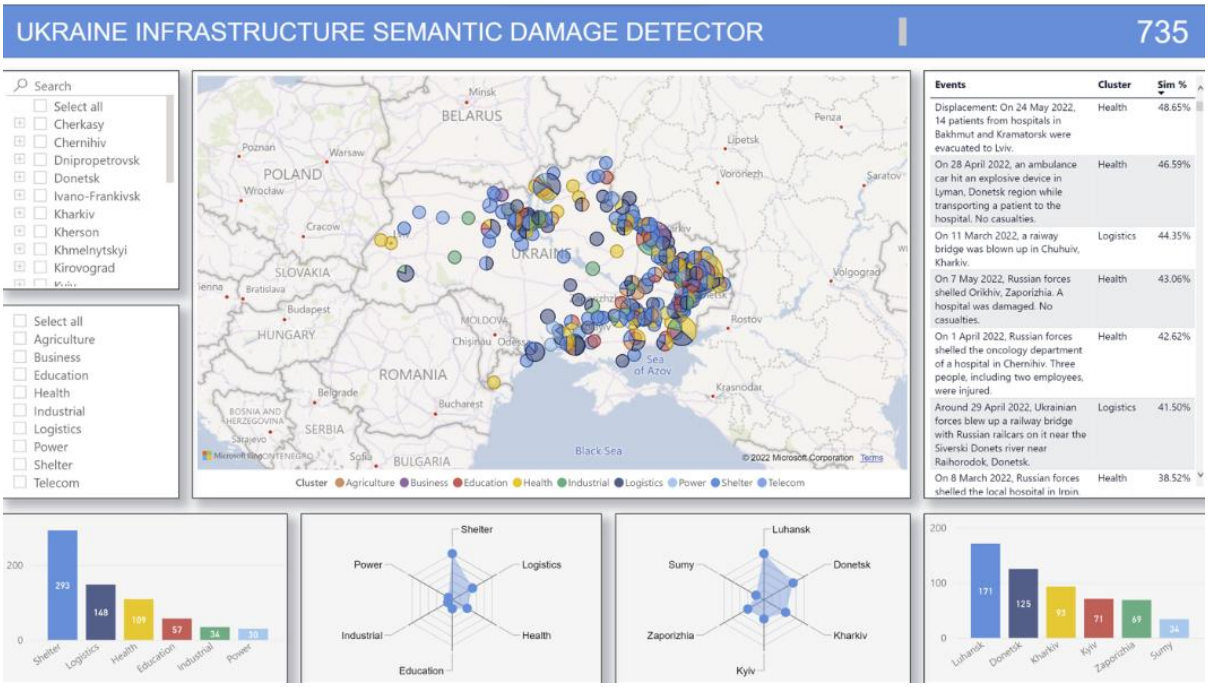


Рисунок 1.2 – Вигляд інтерактивної дошки платформи «Infrastructure Semantic Damage Detector»

Перевагами описаної системи є:

– швидкість та ефективність роботи;

– масштабованість;

– можливість аналізу різних типів інфраструктури.

До недоліків цієї системи можна віднести:

– обмеженість обробкою текстових даних, без аналізу візуальних матеріалів;

– залежність від якості текстових даних без можливості перевірки правдивості отриманих даних даними з інших джерел;

– уклон системи на визначення типу ушкодженої інфраструктури та її розташування та ігнорування питання визначення типу то серйозності пошкоджень та оцінки збитків;

– відсутність порівняння серйозності ушкоджень будівель, особливо при порівнянні ступені пошкоджень інфраструктури одного типу.

Ще одна ініціатива по аналізу та оцінці заданої Україні шкоди через воєнні дії була запущена в лютому 2022 року Інститутом Київської школи економіки разом із владою країни. Ініціативу має назву «Russia will pay» [14]. В її рамках проводиться збір інформації з різних джерел щодо руйнувань інфраструктури України, отриманих під час військових дій, і підрахунку загальних збитків. У рамках проекту оцінюються руйнування різного типу, завдані не тільки будівлям, а й критичній інфраструктурі, дорожнім магістралям, сільськогосподарським угіддям. Дані про руйнування надходять з наявних звітів, наданих органами влади, від бізнес-асоціацій та перевірених ЗМІ. Звичайні громадяни також можуть подати інформацію про руйнування через створені спеціально для цього веб-сайт і Telegram чат-бот, а також через застосунки «Дія» та «Київ Цифровий». Крім того відбувається отримання даних за допомогою супутникових знімків і досліджень звільнених територій дронами.

З квітня 2022 року польсько-американська компанія Tensorflight безоплатно співпрацює з українською владою в рамках описуваної програми. Компанія надає супутникові знімки будівель і за допомогою інструментів машинного навчання та ШІ аналізує їх. За допомогою унікального підходу до отримання зображень (зйомка будівель проводиться зверху, що приблизно відповідає 0° від осі і під кутом у 30 – 50° від осі) у побудованих моделей глибокого навчання є можливість оцінювати ступінь ушкоджень будівель як з пошкодженим дахом, так і споруд, у яких за цілісності даху, є пошкодження фасадів. Після отримання знімків та їхнього попереднього опрацювання відбувається сегментація зображень на неушкоджені та пошкоджені. Далі побудовані моделі розв'язують задачі класифікації, визначаючи ступінь пошкодження пошкоджених будівель: від частково пошкоджених і придатних до відновлення до повністю зруйнованих.

Використовуючи історичні дані про пошкоджену будівлю Tensorflight допомагає попередньо оцінювати вартість реконструкцій. На сьогоднішній

день Tensorflight працювало з виявленням пошкоджень та оцінкою збитків у таких містах, як Буча, Ірпінь, Маріуполь. В рамках програми «Russia will pay» також відбувається допомога прийняття рішень щодо першочерговості відновлення пошкоджених об'єктів. Система має наступні переваги:

– мультимодальний характер отримання інформацій про пошкодження;

– детекція пошкодження фасадів будівель з цілим дахом шляхом використання запропонованого підходу супутникової зйомки.

До недоліків системи, що розглядається можна віднести:

– залежність користувачів (державних установ) від комерційної компанії іншої держави;

– можливість витоку конфіденціальної інформації про об’єкти інфраструктури;

– високу вартість збору та обробки супутникових даних.

Прикладом успішної комерційної компанії, яка займається оцінкою пошкоджень будівель під дією зовнішніх сил, є FlyPix AI. Компанія побудувала платформу геопросторової аналітики – Geospatial AI platform, яка пропонує ряд готових моделей машинного та глибокого навчання, ШІ-інструментів, що дозволяють обробляти знімки з дронів, аерофотозйомки та високоякісні супутникові зображення для автоматизованого виявлення, сегментації та класифікації пошкоджень споруд у реальному часі. Спочатку компанія пропонувала лише моделі для аналізу пошкоджень в результаті стихійних лих, зокрема виявилось дуже корисною при виявлені найбільш ушкоджених населених пунктів та районів після землетрусів в Турції та Японії. У лютому 2025 році компанія повідомила про можливість використання її моделей для аналізу пошкоджень будівель через війну в Україні. При цьому на сайті компанії зазначається, що використання раніше побудованих моделей для виявлення пошкоджень будівель через стихійні лиха не дає високої точності, тому були побудовані нові моделі, що навчанні на користувацьких наборах даних саме пошкодженнями в ході воєнних дій,

а також було використано мультимодальне злиття даних для підвищення точності.

На платформі для сегментації та класифікації пошкоджених будівель представлено моделі машинного та глибокого навчання з різними архітектурами. В своєму блозі [15] компанія повідомляє про те, що для задач сегментації зображень вони використовуються найбільш ефективні моделі глибокого навчання для цієї задачі – U-Net та Mask R-CNN у поєднанні з такими навченими моделями, як ResNet34, SeResNext50, Inception v3 та EfficientNet B4. Найвищу точність вдалось досягти шляхом використання ансамблю моделей Mask R-CNN для пошуку пошкоджених споруд та Inception v3 для оцінки ступеню пошкодженості. Після завершення роботи обраних моделей глибокого навчання система надає користувачам можливість працювати з інформаційними панелями та інтерактивними картами для більш детального ознайомлення з отриманими результатами. До переваг системи, що описується, можна віднести:

– широкий вибір моделей різних типів архітектури з можливістю їхнього до навчання на користувацьких наборах даних для вирішення широкого колу різноманітних задач;

– зручні інструменти для автоматизованому та напів-автоматизованому анотації зображень в користувацькому наборі даних;

– багатофункціональність;

– відсутність необхідності для користувачів мати знання та навички в галузі машинного навчання та штучного інтелекту для користування системою;

– інтеграція налаштованої моделі в хмарні середовища;

– аналіз супутникових і безпілотних знімків в реальному часі.

До недоліків системи можна віднести:

– обмеженість можливостей безкоштовного плану за кількістю запитів до системи, що обмежують можливість використання системи для моніторингу будівель по всій країні;

– відсутність можливості користувачу ознайомитись з детальною інформацією про знайдені пошкодження;

– відсутність інформації про тип споруди, як обстежується;

– відсутність можливості розраховувати пріоритети відновлення будівель з урахування їхньої культурної та соціальної цінності.

Іншою платформою для аналізу пошкодження будівель є платформа від компанії T2D2 – «AI Damage Detector» [16]. Це хмарна платформа, що створена у форматі програмного забезпечення як послуги, яка дозволяє знаходити та класифікувати пошкодження на будівлях за допомогою алгоритмів комп'ютерного зору. Система розроблена для допомоги архітекторам, інспекторам та власникам майна, що повинні стежити за станом будівель. Загалом компанія не ставить собі на меті оцінку пошкодження споруд через воєнні дії, а зосереджує свою увагу на виявлення різних пошкоджень на фасадах, що може частково відноситись до теми дослідження. На рисунку 1.3 наведено вигляд інтерфейсу такої системи.

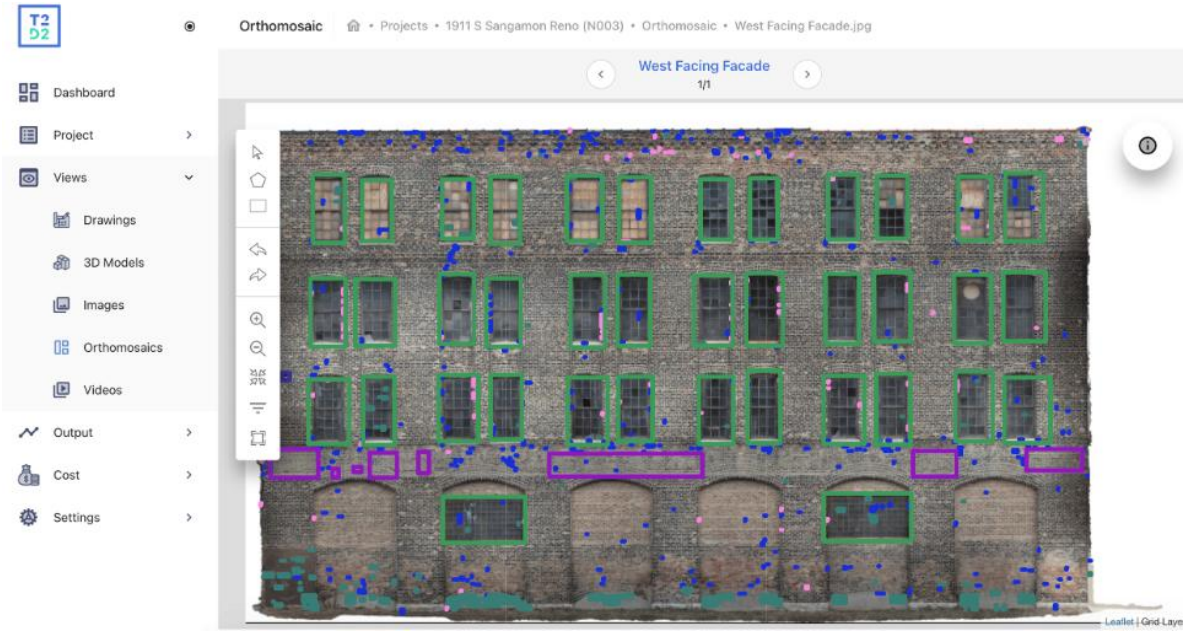


Рисунок 1.3 – Інтерфейс «AI Damage Detector»

Користувачі платформи можуть в якості вхідних даних для

побудованих моделей використовувати зображення та відео, що були отримані з мобільних пристроїв, камер або дронів. Для збору матеріалів з дронів компанія пропонує користуватись послугами професіоналів через свою послугу – T2D2 «Drone Service Provider».

Розробники платформи вказують, що побудовані моделі глибокого навчання здатні розпізнавати понад 80 типів пошкоджень на різних типах будівель. Після завершення обробки наданих матеріалів користувачі можуть отримати звіти з детальними описами знайдених дефектів, з геолокаціями для відстеження місця знаходження інспектованих будівель, інструментами анотації для ручного додавання пошкоджень, 3D-моделі будівель. Крім того, система позволяє відстежувати погіршення стану будівель протягом певного часу, шляхом порівняння зображень та відео отриманих у різні часові проміжки. Розроблене компанією хмарне середовище – T2D2 «Inspection Cloud» дозволяє зберігати потрібні користувачам матеріали та інтегрується з Google Drive, OneDrive, Box.

Переваг цієї системи є:

– можливість отримання детальних звітів з описом знайдених різних типів ушкоджень;

– адаптивність до роботи з різними будівельними матеріалами та широкого спектру руйнувань;

– висока точність класифікації за рахунок великих власних тренувальних наборів даних для навчання запропонованих моделей.

До недоліків такої системи можна віднести:

– відсутність безкоштовного плану (наявний лише двотижневий пробний період);

– не орієнтованість на виявлення пошкоджень, отриманих будівлями в ході воєнний дій, що може погіршити якість виявлення всіх можливих пошкоджень;

– відсутність оцінки значущості будівель у соціально-культурному плані;

– відсутність допомоги в прийнятті рішення про реставрацію споруди а також визначення порідку відновлення будівель.

На початку війни різними волонтерськими організаціями було запущено кілька проєктів, спрямованих на документування руйнувань будівель шляхом побудови їхніх 3D моделей. У таких проєктах розглядаються як прості споруди, так і ті, що становлять культурну або історичну цінність. Як правило, для дослідження споруди використовуються дрони та аерофотозйомка, а створення 3D-моделей відбувається шляхом використання фотограмметричних методів. Отримані результати дають змогу не лише зафіксувати пошкодження, а й використовувати їх для подальшого аналізу фахівцями та планування відновлювальних робіт.

До найвідоміших проєктів можна віднести проєкт «War up Close», який за допомогою панорамних фотографій 360°, кадрів із дронів і 3D-моделювання намагається зафіксувати злочини проти України для подальшого представлення світовій спільноті [17]. На рисинку 1.4 представлено вигляд сторінки проєкту War in 3D з побудованими 3D моделями.

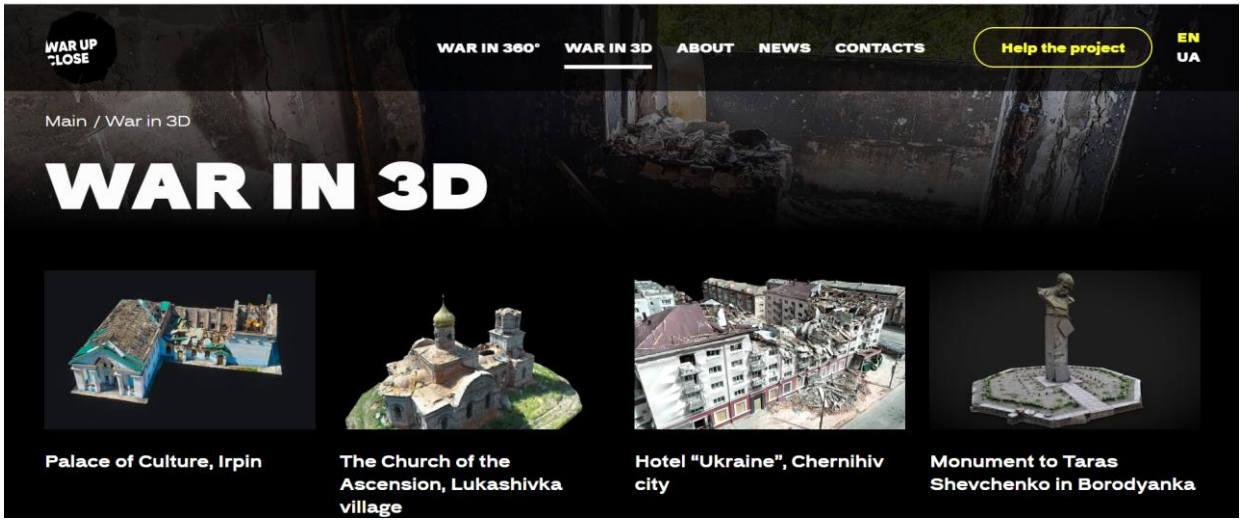


Рисунок 1.4 – Інтерфейс сторінки War in 3D проекту «War up Close»

Команда проєкту активно співпрацює з різними державними службами і допомагає рятувальникам розчищати завали та оцінювати реальний масштаб збитків.

Глобальна ініціатива з відновлення України UNITED24 запустила спільний проєкт з IT-компанією LUN, використовуючи застосунок із запатентованою технологією, який на основі знімків із дронів будує 3D-моделі споруд [18]. Ця ініціатива дає змогу відстежувати динаміку реконструкції пошкоджених будівель і проводить збір коштів на відновлення інфраструктури. На разі дія програми обмежується Києвом та налічує 17 об’єктів. На рисунку 1.5 представлено вигляд інтерактивної карти Києва з відміченими пошкодженими будинками.

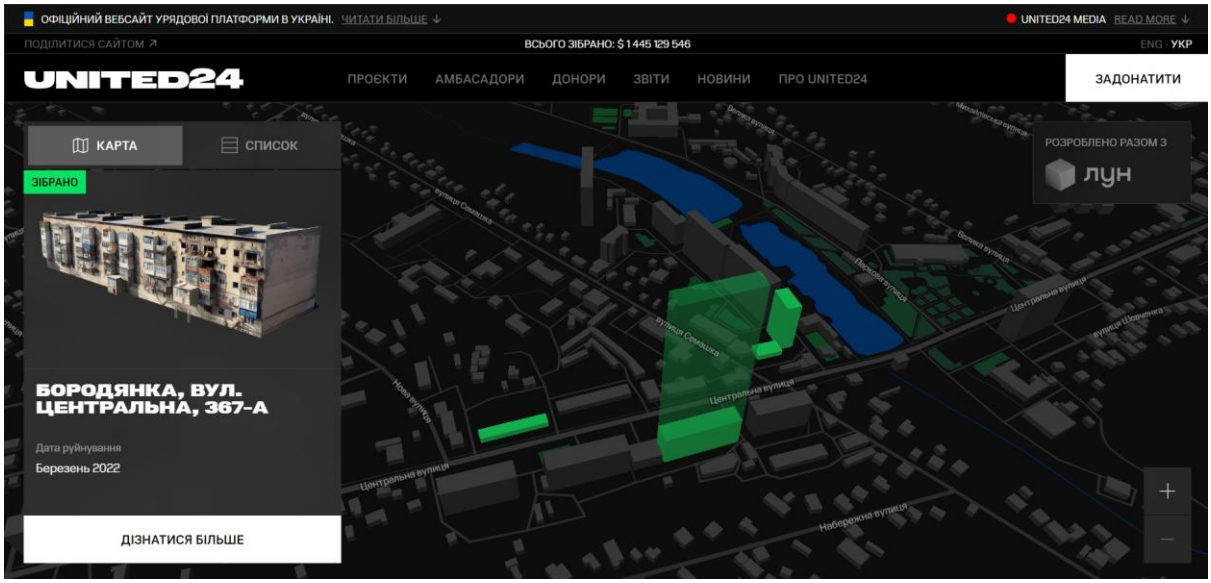


Рисунок 1.5 – Вигляд інтерактивної карти з пошкодженими будинками від UNITED24

На відміну від попередніх ініціатив, які більше зосереджені на простих житлових будинках, волонтерська ініціатива Scan UA працює з об'єктами української спадщини, що були пошкоджені або перебувають під загрозою зникнення [19]. Метою даної ініціативи є створенні цифрового архіву української спадщини для збереження пам'яті про культурну

спадщину, а також для надання інформації про завдану шкоду архітектурі країни.

На рисунку 1.6 представлено створену 3D модель для пошкодженої будівлі пожежної частини № 4 у Харкові.

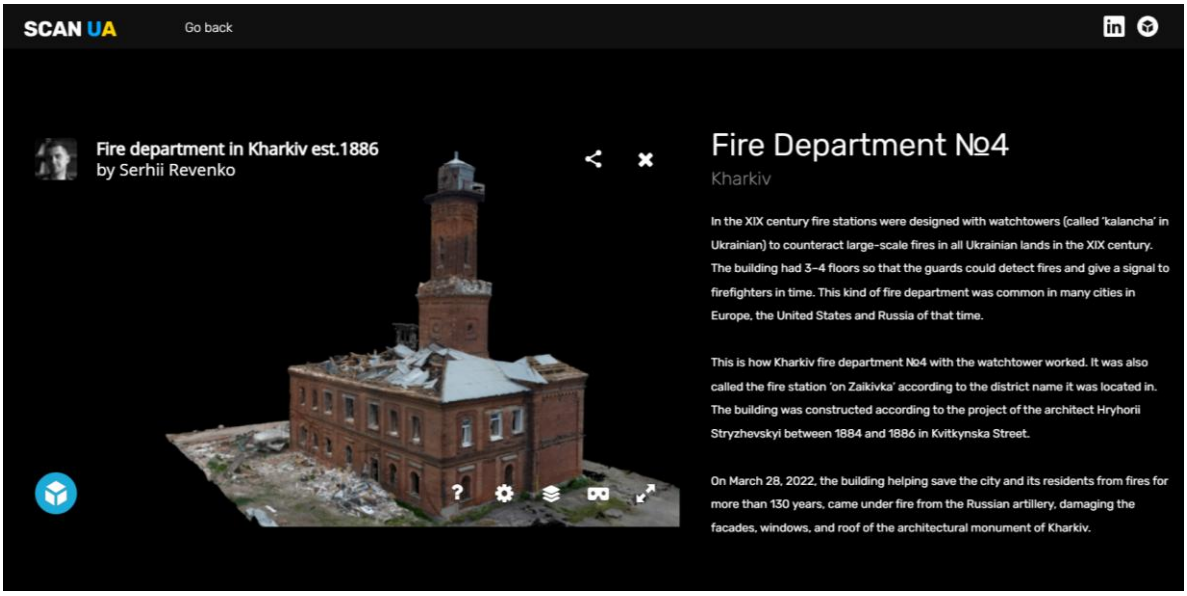


Рисунок 1.6 – Вигляд сторінки присвяченій пожежної частини № 4 у Харкові

Перевагами всіх описаних ініціатив є те, що одержувані в результаті їхньої роботи 3D-моделі дають змогу фахівцям детальніше ознайомитися з присутніми на будівлях ушкодженнями і ретельніше спланувати відновлювальні роботи. При цьому істотним недоліком є відсутність автоматизованого механізму виявлення пошкоджень і визначення рівня зруйнованості будівель. В рамках зазначених ініціатив не пропонується об'єктивного механізму вибору будівель для дослідження на руйнування і ухвалення рішення про черговість проведення відновлювальних робіт.

1.4 Інноваційність та переваги розроблюваної системи

Наведений огляд існуючих платформ та досліджень, які на даний

момент можуть бути застосовані для оцінки ушкоджень будівель в Україні, показав, що не дивлячись на існуючі переваги кожної з них, існують також недоліки та проблеми, що не були вирішені. Основними недоліками систем є їхня залежність від точності одного джерела даних, дорожнеча впровадження, залежність від 3 осіб та передача їм інформації про об'єкти, які знаходяться на території України, відсутність методів об'єктивно-всебічного оцінювання значення споруди та реалізації простих заходів з оцінки пошкоджень будівель. Система, що розробляється в рамках цієї кваліфікаційної роботи, спрямована на подолання зазначених недоліків наявних рішень, пропонуючи інноваційний підхід для оцінювання пошкоджень і пріоритизації відновлення будівель в Україні.

Робота системи буде поділена на 3 основні етапи: виявлення пошкоджених будівель; детальний аналіз виявлених пошкоджених споруд, що включає в себе оцінку цінності кожної з будівель в соціальному та культурному плані, виявлення пошкоджень та оцінка ступеню їх важкості; визначення пріоритетів відновлення враховуючи отримані на попередніх етапах дані.

Однією з важливих переваг розроблюваної системи є можливість отримання даних про пошкодження будівель із кількох джерел, включаючи супутникові дані, звіти органів міської влади та рятувальних служб, інформація від волонтерів та небайдужих громадян. Такий підхід дає змогу отримати більше інформації про наявність пошкоджень споруд, врахувати об’єктів, які могли бути пропущені під час аналізу супутникових знімків. Попри те, що важливу роль при виявленні пошкоджених споруд відіграє робота із супутниковими знімками, яка дасть змогу швидко виявити більшу кількість пошкоджених споруд, вона не є обов'язковою. У разі виникнення проблем із фінансуванням або припиненням підтримки від партнерів , коли використання таких даних стане неможливим, система продовжить функціонувати спираючись на аналіз звітів та інформації про пошкодження від громадян.

Іншою перевагою такої системи є можливість проведення наземного дослідження, що дає змогу більш детально ознайомиться з наявними пошкодженнями. При цьому важливим є те, що система надаватиме не тільки свою загальну оцінку ступеня пошкоджень, а й надаватиме список виявлених пошкоджень. Отримана детальна інформація дає переваги системі і відкриває великі можливості для подальшої роботи. Наприклад, передавши отриманий перелік пошкодження аналітикам і фінансистам з'являється можливість попередньо розрахувати вартість відновлювальних роботи. Маючи перелік пошкоджень з’являється можливість планувати часткову реконструкцію будівель в разі недостатності коштів або оптимізувати оптові закупівлі матеріалів для наборів однотипних пошкоджень у різних будівлях. Іншою корисною ситуацією може бути використання отриманого переліку пошкоджень для ефективного розподілу людських ресурсів. Наприклад, з'являється можливість групувати будівлі за типами пошкоджень, з якими може впоратися певна бригада робітників, складати їхній план роботи та попередньо розраховувати час виконання робіт.

Побудова 3D-моделі будівель дає змогу фахівцям з реконструкцій більш детально ознайомиться з фронтом робіт і вручну ввести пошкодження, які були не помічені моделлю з виявлення ушкоджень, без необхідності попереднього відвідування всіх об'єктів. У ситуації, коли фахівців мало, а пошкоджених будівель багато, це економить час і підвищує ефективність.

При розрахунку пріоритету відновлення система враховує баланс між функціональністю будівлі, ступенем її пошкодження та культурною цінністю, використовуючи стандарти ЮНЕСКО та локальні реєстри. Це забезпечує об’єктивність у виставленні рейтингу, на відміну від FlyPix AI та T2D2, які не враховують культурний аспект, і Russia will pay, де пріоритети визначаються суб’єктивно та враховуються лише соціальне значення споруд.

Нарешті, система є прозорою, надаючи простим громадянам можливість відстежувати, куди спрямовуються їхні кошти, що усуває корупційні ризики. Така відкритість вигідно відрізняє її від інших існуючих застосунків. У контексті триваючого конфлікту в Україні, де ресурси обмежені, а довіра до влади критична, ця перевага забезпечить підтримку громади та донорів, сприяючи швидкому відновленню критичної інфраструктури та культурних об’єктів.

1.5 Постановка задачі

Задача, що ставиться в рамках даної кваліфікаційної роботи, полягає в розробці інтелектуальної системи для автоматизованого виявлення пошкоджень на будівлях за зображеннями, оцінки ступеня ушкоджень та визначення пріоритетності їх відновлення з урахуванням культурного й соціального значення.

Метою даної роботи є розробка комплексного рішення, яке б надало можливості ефективної обробки великих обсягів зображень, ідентифікація рівня руйнувань та прийняття обґрунтованих об’єктивних рішень щодо черговості реконструкції в контексті повоєнного відновлення України.

Для досягнення поставленої мети виконуються наступні кроки:

– створення набору даних, який містять аерофотознімки й наземні фотографії будівель з різними типами пошкоджень;

– збір та структурування інформації про соціальну та культурно-історичну цінність об’єктів;

– розробка моделі комп’ютерного зору для виявлення та класифікації пошкоджень;

– реалізація механізму оцінки ступеня руйнування з урахуванням інформації про проект та серію побудови будівлі;

– створення моделі для оцінки культурної та соціальної значущості об’єктів з використанням інформації отриманої з відкритих джерел,

офіційних баз даних та форумів;

– побудова системи пріоритизації реконструкції будівель, яка здатна враховувати, як серйозність пошкодження будівлі, так і вплив її відновлення на функціонування держави та суспільства;

– розробка зручного інтерфейсу для аналітиків, волонтерів та представників місцевої влади, що дозволяє переглядати наявну інформацію по будівлі, отримані результати роботи моделей, пріоритети відновлення та формувати звіти для планування реконструкції.

Таким чином, основним завданням є створення ефективної, масштабованої та прозорої системи, яка об’єднує штучний інтелект, аналіз зображень та контекстну інформацію для підтримки прийняття рішень щодо реконструкції зруйнованих об’єктів інфраструктури України.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Огляд існуючих досліджень

З метою детального ознайомлення з темою проведення наземного дослідження ушкодження будівель та виявлення ступеню пошкодження була виконана робота по пошуку існуючих наукових робіт та публікацій по даній темі. Далі наведено аналіз джерел, дослідження яких найбільшим чином відповідають темі дослідження даної кваліфікаційної роботи.

Автори наукової роботи [20] запропонували свій підхід автоматизованого виявлення пошкоджених будівель на супутникових і аерокосмічних знімках за допомогою використання методів неконтрольованого машинного навчання.

В якості вхідних даних використовувалися знімки Google Earth міст України до і після початку війни. Перед безпосереднім застосуванням методів машинного навчання проводилося попереднє оброблення зображень, що включало в себе геоприв'язку, ортотрансформацію зображення, підвищення контрастності та вирівнювання яскравості. У такий спосіб значно поліпшувалася якість вхідних даних, що своєю чергою позначалося на точності роботи алгоритмів.

Автори роботи використовували комбінацію таких методів, як Principal Component Analysis (PCA), K-середніх (K-means) та Density-Based Spatial Clustering of Applications with Noise (DBSCAN). На першому етапі алгоритму для виявлення головних векторів, що репрезентують основні напрямки дисперсії даних, використовувався метод PCA. Далі на виявлених векторах застосовувався метод K-середніх, який розділяв зображення на кластери, групуючи схожі області змін. На останньому етапі використовувався DBSCAN для виявлення щільних областей, які вказують на пошкодження. У результаті роботи алгоритму формувалася бінарна маска руйнувань, де пошкоджені ділянки виділялися для подальшого

аналізу. На рисунку 2.1 наведено схему запропонованого авторами статті методу.

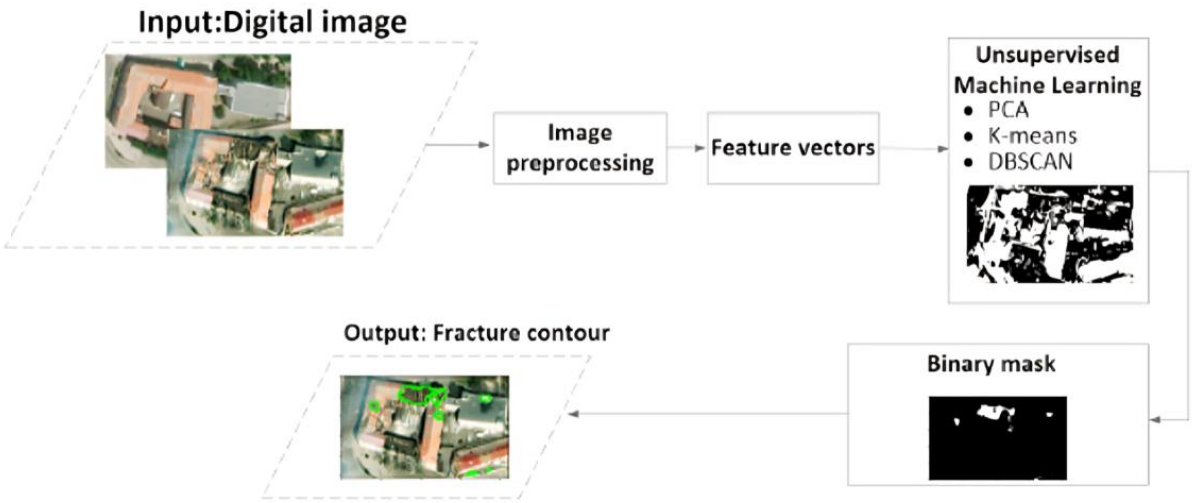


Рисунок 2.1 – Схема запропонованого в статті [20] методу

Як стверджують автори, запропонований метод виявився ефективним і показав високу точність при виявленні пошкоджень на будівлях (Accuracy = 98,13%, F1 Score = 70.90%), що є його безперечною перевагою. Іншою важливою перевагою є можливість алгоритму працювати з даними різної якості та просторової роздільної здатності без необхідності в попередньому глибокому навчанні складних моделей. Як уже згадувалося, цей метод не потребує попередньої розмітки даних. Також використання простих методів машинного навчання дає змогу швидко опрацьовувати зображення та оновлювати карти руйнувань практично в режимі реального часу.

При цьому недоліками запропонованого алгоритму є обмежена здатність до точного визначення меж руйнувань, а також класифікація руйнувань на різні типи. У разі наявності на зображеннях сильних візуальних артефактів, таких як тіні, відблиски або неоднорідне освітлення, точність роботи алгоритму може значно знижуватися. Для використання розробленого методу необхідно мати можливість проведення супутникової

та/або аерокосмічної зйомок, що може бути досить дорогим. При цьому не всі типи пошкоджень можуть бути помітні з висоти. Також отримання необхідних знімків зруйнованої ділянки до і після події не завжди можливе.

Для вирішення зазначених недоліків в наступній науковій роботі [21] авторами було запропоновано новий спосіб визначення ступеню пошкодженості будівель шляхом побудови Damage Detection Map (DDM) та введення нової оцінки – DAV. В якості джерела даних для навчання моделі були використані фото зруйнованих будівель після таких стихійних лих, як землетруси в Непалі та Еквадорі, тайфуну Рубі та урагану Метью. Зйомка будівель проводилась з землі, тому в майбутньому запропонована авторами модель зможе використовуватись для аналізу фото, що були отримані з соціальних мереж, знятими та розмішеними звичайними людьми без використання спеціального обладнання.

В дослідженні для визначення приналежності зображень до класів пошкодження/непошкодження використовується CNN, а точніше використовується модель VGG19 з донавченими зовнішніми шарами на зібраних розмічених наборах даних. До останнього шару згортки застосовується метод CAM, а саме Grad-CAM, який обчислює ваги на основі градієнта для кожної карти об'єктів, виділяючи найбільш релевантні області. Таким чином формується теплова карта DDM, на якій області з високими значеннями інтенсивності відповідають виявленим ділянкам пошкоджень. Для визначення приналежності пошкоджень будівель до незначних, середніх та серйозних пошкоджень використовуються запропонована оцінка DAV, яка розраховується як середнє значення всіх значень на тепловій карті. Чим вище отримане значення, тим важча ступінь пошкодження. Шляхом встановлення порогових значень DAV для виокремлення різних ступенів з’являється можливість підлаштовуватись під різні стихійні лиха, що можуть завдавати різні типи ушкоджень.

В результаті роботи запропонованого методу очікується зображення теплової карти з виокремленими пошкодженнями та клас-ступінь

ушкодженності, до якого було віднесено пошкодження будівлі. На рисунку 2.2 представлено схему описаного методу.

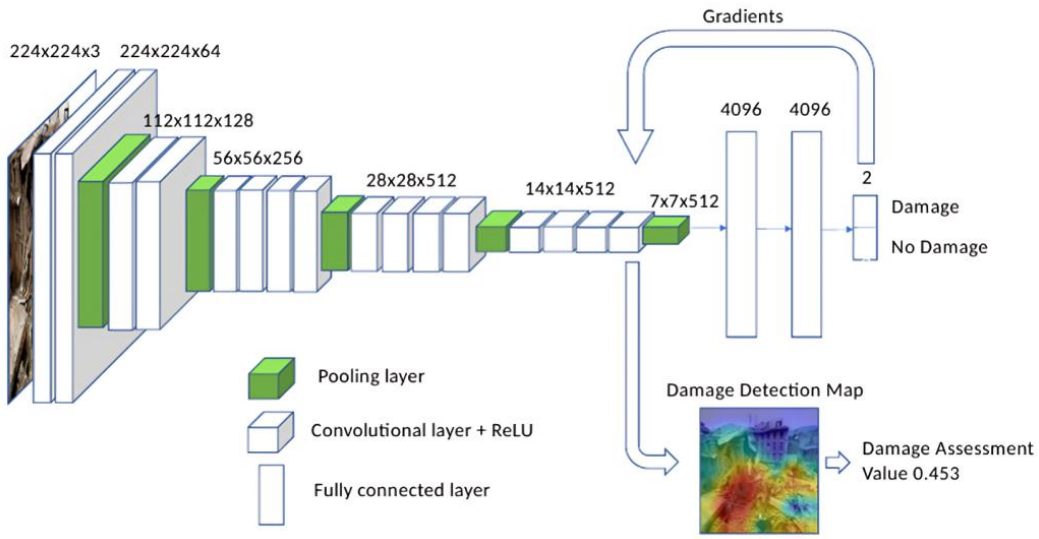


Рисунок 2.2 – Схема запропонованого в статті [21] методу

Перевагою розробленого в статті підходу є його економічність, а також здатність до негайного використання після виникнення надзвичайної ситуації, коли з’являється можливість швидко зібрати великі набори даних з соціальних мереж. Іншою перевагою є наочність та зрозумілість отриманих результатів, що дозволяє підвищити довіру до автоматизованих систем виявлення та класифікації ступеню ушкоджень. У продовженні роботи за цією темою дослідники випустили статтю [22], яка показує переваги використання Grad-CAM and Grad-CAM++ у розробленому методі, а також демонструє особливості роботи моделі для класифікації ушкоджень інфраструктури різних типів, як-от дороги, мости та будівлі.

Недоліком такої роботи може бути те, що деякі руйнування, які є досить маленькими за обсягом, не будуть виділені на теплових картах, а отже можуть бути не враховані фахівцями під час проведення відновлювальних робіт. Запропонований метод може бути використаний

для візуалізації великих пошкоджень будівель, але він включає алгоритм автоматичної оцінки важкості пошкодження будівлі.

В наступній роботі [23] також було запропоновано метод, який працює з зображеннями, отриманими шляхом наземної зйомки та не потребує наявності фото будівель до та після руйнування, однак у порівнянні з попередньо описаними робота він має перевагу, що полягає в визначенні ступеню пошкодження будівель як результат своєї роботи.

Автори статті описують підхід створення комплексної складної системи для обчислення оцінки пошкодженості споруди за шкалою від 0 до 1. В якості джерела даних для тренування моделей було створено власний датасет, що складається з 250 зображень будівель після стихійних лих, зібраних з відкритих джерел, і вручну попередньо його анотовано.

Для виявлення ознак, які на фінальному етапі подаються регрессору, що передбачає серйозність ушкодження будівлі, було розроблено три окремі потоки обробки ознак. Потік виявлення ознак кольорового зображення працює із зображеннями, які надходять на вхід, без попередньої обробки. Для виявлення глибинних ознак використовується конволюційна нейронна мережа архітектури VGG. У потоці виявлення ознак кольорової маски першим етапом проводиться семантична сегментація зображення за допомогою DilatedNet, під час якої забираються зайві об'єкти, що могли випадково потрапити на зображення, та залишаються лише виокремлювати тільки релевантні об'єкти, такі як самі будівлі, їхні стіни. Наступний етап збігається з потоком виявлення ознак кольорового зображення, відрізняючись тільки даними, які подаються на його вхід. Останнім є потік виявлення ознак бінарної маски. Після очищення зображення від зайвих об'єктів за допомогою DilatedNet, зображення перетворюється на чорно-білу маску, в якій представлено контур споруди з пошкодженими ділянками. Виявлення ознак відбувалось з використанням полегшеної архітектури LeNet.

Отримані ознаки з кожного потоку об'єднуються в єдиний вектор ознак, який потім обробляється повнозв'язним шаром із сигмоїдальною функцією активації для визначення рівня руйнування будівлі. На рисунку 2.3 представлено блок-схему для більш повного розуміння описаного підходу.

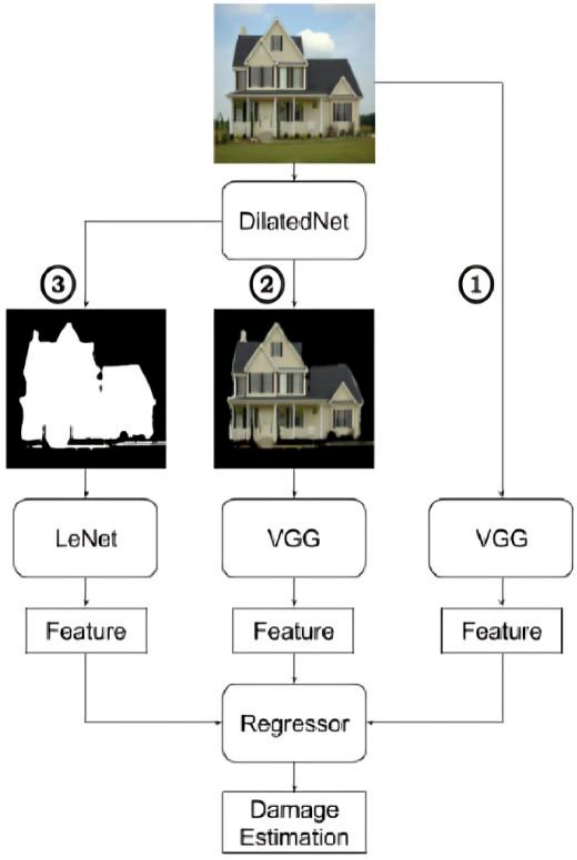


Рисунок 2.3 – Схема запропонованого в статті [23] методу

Перевагами описаного підходу є можливість застосування його для об'єктів без наявності інформації про його стан до руйнування. Також розроблена модель забезпечує оцінку ступеня пошкодження у вигляді безперервної величини, що дає змогу використовувати отриманий результат у визначенні пріоритетів відновлення споруд. Виявлення ознак 3 різними способами дає змогу працювати з різними типами ушкоджень і збільшити точність передбачення регрессора.

Водночас складність моделі є і недоліком такого підходу. Варто очікувати на збільшення потреби в обчислювальних ресурсах і часу на навчання моделі. Іноді для більш детального планування відновлювальних робіт і визначення пріоритетів відновлення може бути недостатнім оцінки серйозності пошкодження, а потрібен перелік наявних пошкоджень. Також оцінка серйозності руйнування будівлі залежить від тренувального набору даних, які були попередньо проанотовані певною групою експертів. Розбіжність у оцінці ступеня пошкодження різних груп експертів або зміна вимог до критеріїв визначення тяжкості пошкоджень призведе до необхідності підготовки нового набору даних і повторного навчання моделі.

Вирішенню проблем, пов'язаних зі складністю моделі, може сприяти використання підходу, запропонованого як частина проєкту, описаного в статті [24]. Автор цієї роботи використовує методи комп'ютерного зору для оцінювання ступеня пошкодження будівель і визначення типового проєкту, за яким будували цю будівлю, що надає інформацію про будівельний матеріал, який використовували під час зведення будівель.

Джерелом даних стала користувацька карта Google My Maps, створена добровольцем у квітні 2022 року яка містить зображення пошкоджених будівель в місті Маріуполь, як з наземного ракурсу, так і з дронів. В рамках роботи було проведено анотування отриманих зображень за допомогою платформи Roboflow за чотирма класами ступеню ушкодження будівлі, адаптованими з класифікації EMS 98. Для збільшення кількості зображень, а також підвищення стійкості моделі, що розроблюється були використані різні методи аугментації даних. Для автоматичного виявлення рівня пошкодження будівлі використовувалась донавчена модель YOLOv8. Як зазначає автор статті навчання моделі проводилось протягом 50 епох і показало такі результати: mAP 58,4 %, Precision 64,1 %, Recall 52,5 %. Для визначення типу проекту забудови також була використана зазначена архітектура моделі глибокого навчання.

Перевагою цього проєкту є використання в роботі відкритого джерела даних, створеного очевидцями. Зображення, на яких навчалася модель, різноманітні за якістю, відстанню до будівлі, ракурсами. Зйомка будівель велася як із землі, так і з використанням дронів. Такий підхід гарантує стабільність моделі під час роботи в майбутньому з новими зображеннями, які можуть бути різної якості та виконані без попереднього узгодження щодо розміщення споруди в кадрі. Також перевагою даного дослідження, є використання сучасної універсальної моделі – Yolov8, що дасть змогу в майбутньому забезпечити баланс між точністю та продуктивністю передбачень.

До недоліків цієї роботи можна віднести обраний підхід класифікації ступеня пошкодженості будівлі. Для визначення серйозності пошкодження будівлі використовували 5-бальну шкалу оцінки пошкодження будівлі після землетрусу. Пошкодження будівлі після землетрусу можуть відрізнятися від пошкоджень, отриманих будівлею під час військових дій, що може позначитися на точності роботи моделі. Також суттєвим недоліком є суб'єктивність віднесення споруди на кожному зображенні до кожного з класів.

Рішенням описаних проблем може бути підхід, запропонований у роботі [25]. У рамках цієї наукової роботи проводилося дослідження пошкоджень будівель на наземних зображеннях за допомогою полегшеної версії моделі YOLOv5. Основною метою цього дослідження, яку ставили її автори, було розроблення системи, здатної працювати в реальному часі, зберігаючи високу точність за мінімальних обчислювальних витрат. Для навчання і тестування методу, що розробляється, було створено спеціальний датасет Ground-level Detection in Building Damage Assessment, що складався з наземних фотографій пошкоджених будівель та анотацій, що включають різні типи пошкоджень. Модель навчали розпізнавати такі типи руйнувань, як обвали, тріщини, відшарування матеріалів і купу уламків. Під час розроблення власної моделі LA-YOLOv5, для YOLOv5, взяту за основу,

було проведено низку змін, а саме було додано елементи Ghost Bottleneck та CBAM (Convolutional Block Attention Module), а також покращено обробку масштабів об'єктів. У результаті було отримано модель, яка зберегла високу точність, за значного зменшення обсягу пам'яті, який потрібен для її розгортання.

Перевагами описаного в роботі підходу є висока швидкість роботи моделі, що дає змогу використовувати її в ситуаціях, які потребують виявлення пошкоджень практично в реальному часі та швидкого реагування. Іншою важливою перевагою є можливість виявлення різних типів пошкоджень будівель, що дає змогу глибше аналізувати руйнування та ухвалювати рішення щодо їхньої пріоритезації під час відновлювальних робіт.

Недоліками цієї роботи є використання не останньої версії архітектури YOLO, а також відсутність алгоритму оцінки ступеня пошкодження будівель.

Під час ознайомлення з низкою наукових праць, присвячених виявленню ушкоджень будівель та оцінці серйозності руйнувань, були розглянуті різні підходи до розв'язання поставленої задачі, кожен з яких має як свої переваги, так і недоліки. Основним виявленим недоліком було визначено відсутність об'єктивного підходу до оцінювання ступеня пошкодження будівлі та можливості адаптації цього способу до різних ситуацій, які потребують застосування розроблених моделей.

Для вивчення способів розв'язання задачі визначення важливості відновлення однієї споруди над іншою було проведено дослідження наявних наукових робіт, що описують способи розв'язання схожих проблем. У статті [26] описується застосування MCDA для визначення пріоритетних об'єктів інфраструктури для будівництва або проведення реконструкцій в умовах обмеженості фінансів. Автори цієї роботи поставили перед собою завдання виявлення доріг, які найбільше за все потребують проведення ремонтних робіт у м. Рієці (Хорватія). Після складання переліку доріг, які

варто розглянути, та вибору критеріїв, за якими проводитиметься порівняння доріг, у науковій роботі описується розроблений спосіб ухвалення рішення, що складається з комбінації двофазного AHP та методу TOPSIS. Для визначення ваг обраних критеріїв було використано метод AHP, який як зазначалося складається з двох фаз. На першій фазі ваги критеріїв розставлялись з урахуванням думки всіх зацікавлених у проведенні ремонтних робіт сторін, на другій фазі відбувалось коригування визначених ваг на підставі рішення головного експерта (ключової особи). Далі виконувався розрахунок пріоритетності для кожного об'єкта методом TOPSIS шляхом ранжування об'єктів за їхньою близькістю до ідеального об'єкта. Таким чином у результаті роботи цього методу було сформовано список пріоритетних для відновлення доріг та їхніх ділянок.

Явною перевагою такого підхід розробленого методу є врахування думок усіх зацікавлених сторін під час визначення ваг критеріїв, що дає змогу врахувати різні точки зору на виконання поставленого завдання. Сам процес ранжування є простим для розуміння і робить процес визначення пріоритету одного об'єкта над іншим прозорим та об'єктивним. Недоліком запропонованого методу є як часова, так і обчислювальна складність налаштування методу AHP, особливо під час розгляду великої кількості критеріїв та за участю великої кількості експертів.

Для вирішення зазначеної проблеми слід розглянути інший підхід запропонований в науковій роботі [27], де автори реалізували комплексний підхід для визначення черговості реалізації запропонованих проєктів відновлення після стихійних лих.

У даній роботі для кожного проєкту визначався набір критеріїв, за якими відбувалось їх порівняння. Для визначення важливості кожного критерію використовувався метод Best Worst Method (BWM), що дав змогу експертам обрати найбільш і найменш важливі критерії для проєктів і на підставі їхніх порівнянь розрахувати ваги для всієї системи оцінювання. Далі в роботі було застосовано 4 MCDM методи ранжування: TOPSIS,

ELECTRE III, VIKOR і PROMETHEE. Для отримання фінального результату (списку проєктів) було використано метод лінійного присвоювання для об'єднання результатів, наданих кожним з методів ранжування, у єдиний вектор пріоритетів з мінімізацією розбіжностей.

Для перевірки достовірності отриманих результатів автори використали ANN, натреновану на історичних даних про реалізовані проєкти. Були зроблені висновки про те, що отримані методом MCDA результати збігаються з прогнозами, які дала нейрона мережа. Таким чином було підтверджено надійність розробленого підходу.

Перевагами описаного методу є використання чотирьох MCDM методів, що дозволило отримати найбільш точні результати ранжування. Іншою перевагу способу є використання методу BWM для визначення ваг критеріїв, який є менш трудомістким у порівняні з методом AHP, але також забезпечує високу узгодженість рішень.

Не зважаючи на те, що не було знайдено наукових робіт присвячених ранжуванню та визначенню пріоритетності відновлення одній будівлі над іншій з урахуванням ступеню їх пошкодженності, соціального та культурного значень, розглянуті способи можуть бути використані з незначними змінами для вирішення поставленої в рамках даної кваліфікаційної роботи задачі.

2.2 Огляд та порівняння напрямків комп’ютерного зору

Сучасний стан розвитку комп’ютерного зору характеризується здатністю вирішувати широкий спектр завдань, які знаходять своє застосування в різноманітних галузях людської діяльності. Згідно з даними, наданими в статті [28], у 2025 році найпопулярнішими завданнями комп'ютерного зору повинні стати класифікація зображень (image classification), виявлення об'єктів (object detection), семантична сегментація (semantic segmentation), сегментація екземплярів (instance

segmentation), оцінка пози (pose estimation) і генерація зображень (image generation).

Перед початком розробки системи комп’ютерного зору дуже важливим є чітке визначення типу задачі, що потребує вирішення. Правильно зроблене рішення дозволяє зрозуміти в якому вигляді потрібно очікувати результати роботи системи, обрати оптимальну модель, яка буде відповідати умовам функціонування системи, визначити метрики для оцінки точності, а також розробити або налаштувати алгоритм для подальшої обробки отриманих результатів.

У контексті задачі виявлення руйнувань будівель в ході воєнних дій та оцінки ступеню їхньої пошкодженості, є необхідним обрання методу комп’ютерного зору, який може забезпечити ефективне виявлення, класифікацію та аналіз пошкоджень на вхідних зображеннях. Для вирішення цього завдання можуть бути застосовані такі підходи, як класифікація зображень, виявлення об'єктів, семантична сегментація та сегментація екземплярів. На рисунку 2.1 показані візуальні відмінності між зазначеними підходами, що представлені в науковій роботі [29]. Далі наводяться їхній детальний огляд, аналіз переваг та недоліків, а також обґрунтовується вибір оптимального підходу для вирішення поставленої в рамках даної роботи задачі.

Одним з фундаментальних напрямків комп’ютерного зору є класифікація зображень, що передбачає присвоєння зображенню однієї або кількох категорій на основі його візуальних характеристик [30]. Моделі класифікації навчаються на наборі даних із попередньо розміченими зображеннями, де кожне зображення повинно відповідати певному класу. В рамках задачі, що вирішується в даній кваліфікаційній роботі, це можуть бути зображення з текстовими позначками «пошкоджена будівля», «частково пошкоджена» або «непошкоджена» чи числовими значеннями – ступенями серйозності руйнування будівлі.

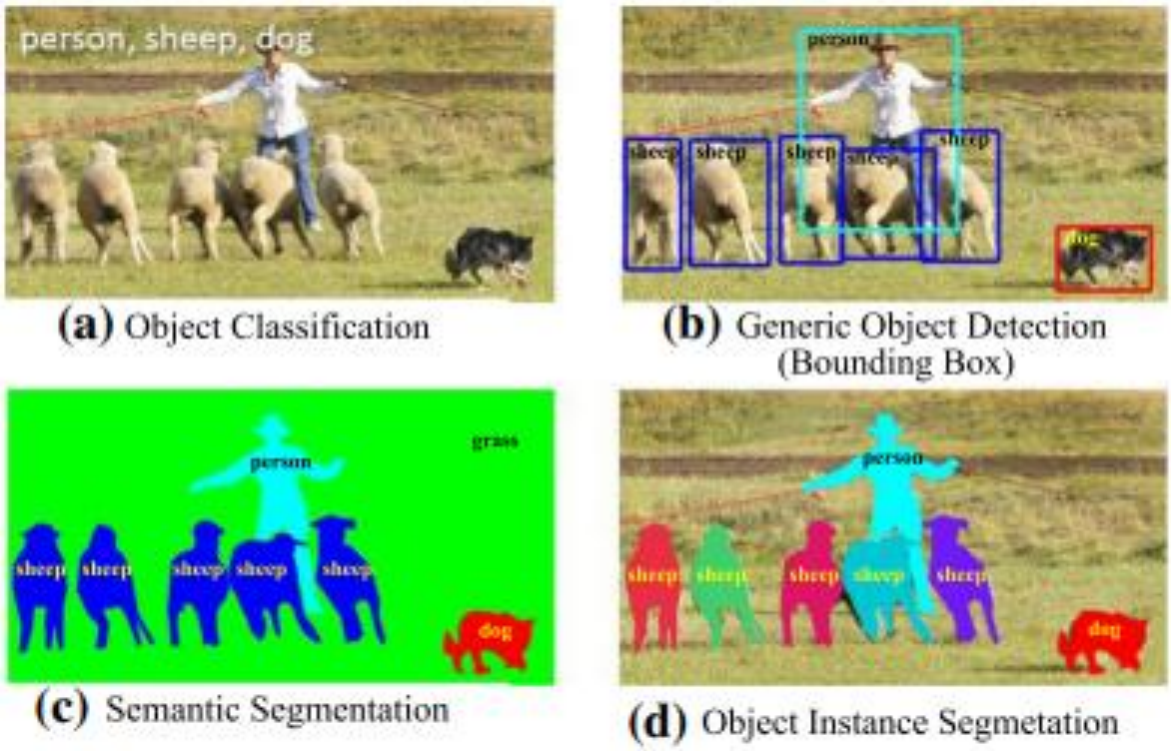


Рисунок 2.4 – Основні напрямки комп’ютерного зору

Перевагою цього напрямку є його відносна простота реалізації, що частково зумовлена великою кількістю вже натренованих моделей, які добре працюють з класифікацією, таких як ResNet або VGG, що потребують мінімального донавчення для демонстрації високої точності [31]. Іншими перевагами є висока швидкість обробки зображення та потребою в невеликих обчислювальних ресурсів порівняно з роботою методів в інших напрямках.

Водночас цей напрямок класифікації зображень має суттєві недоліки: він не здатен визначити розташування чи межі пошкоджень, що значно обмежує його інформативність для задач, які потребують локалізації. Одним із ключових недоліків є необхідність залучення експертів для анотації даних, що може призводити до суб’єктивності в оцінках, оскільки різні фахівці можуть по-різному інтерпретувати ступінь пошкодження будівель під час візуального аналізу. Це створює ризик неоднорідності міток, що негативно впливає на надійність моделі. Таким чином, хоча цей

метод не слід повністю виключати з розгляду, доцільно звернути увагу на інші підходи комп'ютерного зору, які забезпечують локалізацію та детальний аналіз пошкоджень, дозволяючи відокремити процес оцінки ступеня руйнувань від моделі та винести його на наступний етап.

Іншим важливим напрямом комп'ютерного зору є детекція об'єктів, який зосереджується на ідентифікації та локалізації об'єктів на зображенні шляхом створення обмежувальних рамок навколо них із подальшою класифікацією. У процесі детекції модель навчається розпізнавати об'єкти на основі їхніх візуальних характеристик, таких як форма, текстура чи колір, і створювати прямокутні рамки, які охоплюють ці об'єкти, разом із мітками, що вказують на їхній клас [32]. Його особливістю є те, що він ідентифікує об'єкти як окремі одиниці, але не надає інформації про їхні точні межі.

Головною перевагою такого підходу є те, що він дозволяє точно визначити місце розташування об'єктів на зображенні, що є значним прогресом порівняно з класифікацією зображень. При цьому, його недоліками є те, що обмежувальні рамки не відображають точну форму чи межі об'єктів, що знижує точність аналізу в задачах, де потрібна детальна інформація про структуру пошкоджень. У випадках, коли об'єкти перекриваються, мають нечіткі межі або розташовані щільно один до одного, даний підхід може втрачати точність, оскільки рамки можуть неправильно охоплювати об'єкти або пропускати їх [33].

Отже, використання такого підходу для виявлення пошкоджень будівель є можливим, однак це не дозволить отримати детальну інформацію про форму чи точні межі пошкоджень.

Для досягнення найбільшої ефективності варто розглянути напрямок сегментації зображень, який дозволяє не тільки визначити місце знаходження об'єктів але й його точні контури. Окрім того, одним із суттєвих переваг цього напрямку комп'ютерного зору є його багатогранність, що проявляється у широкому спектрі підходів, які можна ефективно застосовувати – від сучасних глибоких нейронних мереж до

традиційних ланцюгів Маркова [34]. Як вже зазначалось раніше, в цьому напрямку існує два основні під напрямки – семантична сегментація та сегментація екземплярів, які детально розглянуті в [35].

Семантична сегментація передбачає класифікацію кожного пікселя зображення, визначаючи його належність до певного класу. Особливістю даного напрямку є те, що не розрізняються окремі екземпляри одного класу, а відносяться та позначаються, як одна категорія. Через те, що метод забезпечує детальний аналіз зображення на рівні пікселів, з'являється можливість точно оцінити площу та розподіл об'єктів на зображені. Крім того, семантична сегментація добре справляється з аналізом зображень із великою кількістю об'єктів або складною структурою, де необхідна сегментація всієї сцени.

Отримання в результаті використання методів такого напрямку детальної інформації про об'єкти та їхні чіткі межі і є головним недоліком даного підходу. При цьому за недолік даного підходу можна вважати неможливість розрізнення окремих екземплярів одного класу, що обмежує його використання в задачах, де потрібна ідентифікація окремих об'єктів. Для задачі, що розглядається, використання семантичної сегментація дозволить оцінити загальну площу пошкоджень, однак не зможе надати інформацію про кількість або форму окремих пошкоджень одного типу.

Виправити зазначений недолік може сегментація екземплярів, яка поєднує можливості детекції об'єктів і семантичної сегментації, забезпечуючи ідентифікацію, локалізацію та піксельну класифікацію окремих екземплярів об'єктів на зображенні. Даний метод не лише визначає категорію об'єктів, а й створює окремі піксельні маски для кожного екземпляра, дозволяючи чітко відокремлювати різні об'єкти одного класу. Перевагами такого підхіду є можливість повного детального аналізу зображень із великою кількістю об'єктів або складною структурою, з наданням інформації по всім виявленим елементам. До можливих недоліків

такого підходу можна віднести необхідність в більших обчислювальних ресурсів, у порівнянні з іншими напрямами.

Отже, для задачі виявлення пошкоджень на будівлях найкращий є використання сегментації екземплярів, що дозволяє не лише виявити пошкодження, а й точно визначити їх межі та кількість, що є важливим для всебічної оцінки масштабу руйнувань та планування заходів відновлення.

2.3 Порівняння моделей сегментації екземплярів

Як було визначено в попередньому пункті сегментація екземплярів є оптимальним підходом для вирішення задачі виявлення пошкоджень будівель, спричинених воєнними діями, де на одному зображенні може бути від 10 до 100 дрібних пошкоджень із сильним дисбалансом класів. Після визначення напряму комп'ютерного зору важливим стає правильний вибір моделі з цього напряму, яка здатна точно визначати всі пошкодження та мати високу швидкість роботи. Цей розділ присвячено огляду сучасних моделей комп'ютерного зору для сегментації екземплярів з поясненням їхньої придатності для вирішення поставленої задачі.

За результатами проведення порівняльного аналізу різних джерел було виявлено відсутність єдиної думки щодо найкращої моделі сегментації об'єктів, враховуючи різноманітність критеріїв порівняння: точність сегментації, швидкість обробки, обчислювальна ефективність і адаптивність до дисбалансу класів. Так, у статті [36] представлено огляд моделей сегментації екземплярів, що вважались кращими у 2024 році. Автори роботи наводять три рейтинги моделей, порівнюючи їх за такими ознаками: швидкість роботи, точність та можливість налаштування через промпти. У роботі порівняння моделей проводяться за оцінками mAP для обмежувальних рамок, масок, а також зручності налаштування та підтримкою спільноти. Першими порівнюються моделі сегментації, що здатні працювати в режимі реального часу. Виокремлюються такі моделі, як

моделі сімейства YOLO – YOLOv8-seg та YOLOv9-seg, які мають значення оцінок mAP для обмежувальних рамок та масок (36.7 – 53.4, 20.5 – 43.4) та (53.3, 43.5) відповідно. Незважаючи на те, що восьма версія моделі має більшу підтримку спільноти та простіше налаштування, як показано вище більш сучасна дев'ята версія показує стабільніші та вищі результати. Іншими моделями, що розглядаються, є RTMDet-Ins, YOLACT++, SparseInst і Detectron2, які також добре оптимізовані для роботи в режимі реального часу, однак трохи поступаються в точності моделям YOLO. Враховуючи, що останні версії моделей сімейства YOLO мають стабільно вищі оцінки значення mAP: 54.7 для обмежувальних рамок та 43.8 масок [37], інші моделі значно відстають.

В якості кращих моделей, що є більш повільними, але забезпечують високу точність визначення об'єктів та увагу до маленьких деталей, в статті наводяться сучасні архітектури моделей BEiT3, Mask2Former, MaskDINO, OneFormer, які показують великі значення метрики mAP для масок: 54.8, 50.5, 52.3, 49.2 відповідно. Однак їхнім суттєвим недоліком, що ускладнює їхнє використання на практиці і, особливо в рамках даної кваліфікаційної роботи, є висока обчислювальна складність, що вимагає значних апаратних ресурсів, недоступних для звичайних локальних комп'ютерів. Крім того, ці моделі потребують великих обсягів детально анотованих даних для донавчання, що є трудомістким і з погляду на збір потрібної кількості зображень, і їхньої анотації.

Останнім рейтингом моделей, що наводиться в рамках статті, є порівняння моделей сегментації екземплярів з можливістю налаштування за допомогою промтів. Хоча такі моделі, як SAM, MobileSAM та Grounded- SAM можуть виявитись дуже корисними при анотуванні зображень при підготовці набору даних, вони є непридатними для використання для поставленої задачі в рамках цієї кваліфікаційної роботи.

В іншій оглядовій статті [38] в якості найактуальніших моделей сегментації згадуються такі моделі, як YOLOv8, Mask R-CNN, Panoptic FPN

та YOLACT++. В статті зазначається, що YOLOv8 виділяється високою швидкістю і можливостями для обробки зображень в режимі реального часу, що робить його ідеальним для таких застосунків, як автономне водіння і системи відеоспостереження. Mask R-CNN забезпечує точну піксельну сегментацію і підходить для завдань, що вимагають максимальної точності, наприклад, у медичній візуалізації. Panoptic FPN поєднує сегментацію екземплярів і семантичну сегментацію, надаючи комплексне розуміння сцени, що актуально для робототехніки та автономної навігації. YOLACT++ орієнтований на швидку сегментацію екземплярів у реальному часі, що корисно для відеоаналізу, розпізнавання дій і підрахунку об'єктів. При цьому автори зазначають недоцільність порівняння моделей відірвано від їхнього використання на практиці. Переваги той чи іншої моделі повинні надаватись враховуючи проблеми, що постають у кожній конкретній задачі та можливості моделей її вирішувати.

Отже, на даному етапі можна визначити, що для вирішення завдання виявлення пошкоджень будівель серед розглянутих моделей варто звернути особливу увагу на моделі сімейства YOLO і Mask R-CNN, враховуючи, що перші здатна поєднувати високу швидкість обробки зображень з хорошою точністю, а друга зберігає лідерство в завданнях, де важлива максимальна точність піксельної сегментації та деталізація.

2.4 Огляд моделей YOLOv11 та Mask-R-CNN

Моделі YOLOv11 і Mask R-CNN представляють собою два різні підходи до вирішення завдання сегментації екземплярів, кожна з яких має свої архітектурні особливості, переваги та обмеження.

Модель YOLOv11 являє собою сучасний одноетапний алгоритм для задач сегментації об'єктів, побудований на базі архітектури сімейства YOLO, яку від самого початку розробили для швидкої та ефективної детекції об'єктів у зображеннях, а згодом розвинули для виконання задач

сегментації [39]. Основний принцип роботи таких моделей полягає в поділі зображення на сітку й одночасному передбаченні обмежувальних рамок, класів об'єктів і сегментаційних масок для кожного регіону в межах єдиного проходу по зображенню. На рисунку 2.5 представлено схему архітектури такої моделі [40].

Архітектура YOLOv11 включає поліпшений backbone, заснований на Darknet-53, який забезпечує ефективне вилучення ознак за допомогою згорткових шарів. У моделі застосовується модифікований C2F модуль, що підвищує якість вилучення детальних ознак, а також зменшено розміри згорткових фільтрів із 6×6 до 3×3, що знижує обчислювальні витрати. Значущою особливістю є використання роздільної голови (decoupled head), яка розділяє задачі регресії координат і класифікації, покращуючи продуктивність і точність сегментації.

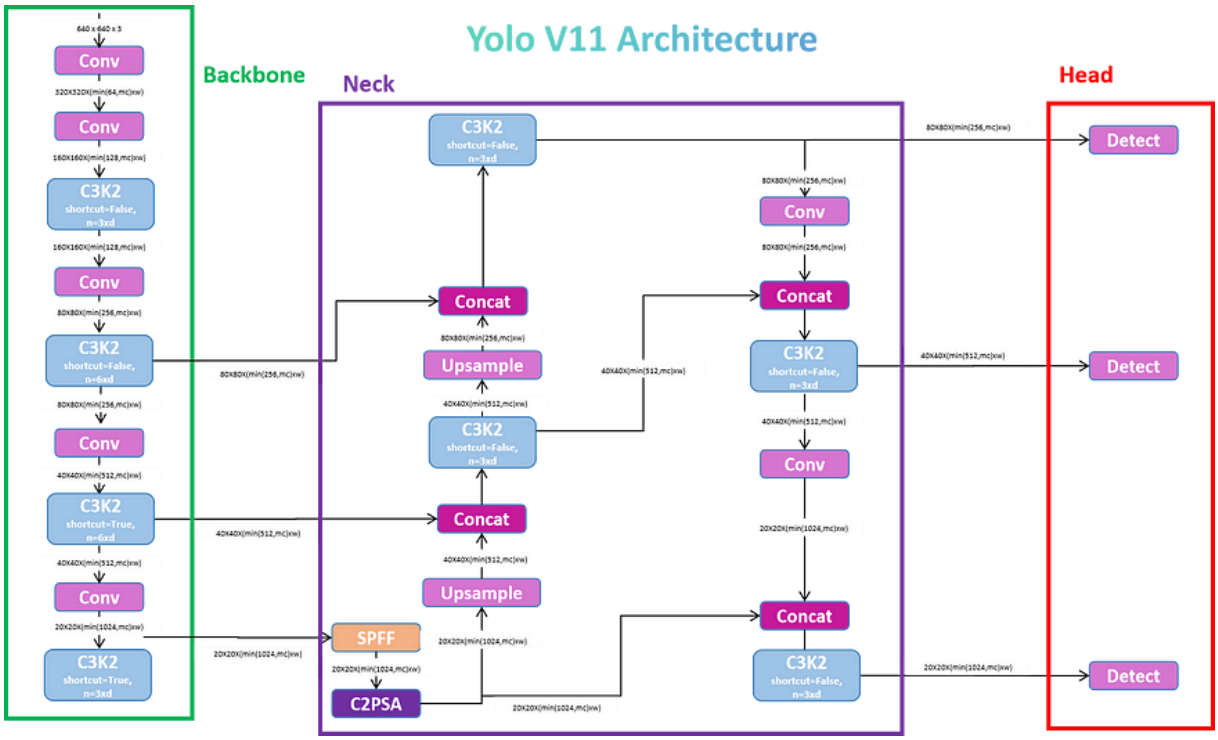


Рисунок 2.5 – Архітектура YOLOv11

Для підвищення стійкості та якості прогнозів використовується метод

Pseudo Ensemble, що дає змогу об'єднувати висновки кількох моделей для більш стабільного результату. Ці архітектурні рішення роблять YOLOv11 дуже швидкою та ефективною моделлю, здатною обробляти до 128 кадрів за секунду, що особливо важливо для застосунків у реальному часі та систем з обмеженими ресурсами.

Основними перевагами YOLOv11 є висока швидкість інференсу і збалансована точність, що дає змогу застосовувати модель у мобільних і вбудованих пристроях. Однак, незважаючи на поліпшену архітектуру, модель може демонструвати невелике зниження точності у порівняні з двоетапними методами, особливо під час роботи зі складними або дрібними об'єктами.

На противагу YOLOv11, Mask R-CNN є двоетапною моделлю, яка досягла широкого визнання завдяки високій точності сегментації об'єктів [41]. Архітектура Mask R-CNN включає backbone-мережу (зазвичай ResNet-50 або ResNet-101 з FPN), яка витягує багаторівневі ознаки із зображення. Далі Region Proposal Network (RPN) генерує регіональні пропозиції, які за допомогою RoI Align шару точно вирівнюються для вилучення ознак конкретних регіонів інтересу. Масштабні ознаки подаються в голову моделі, яка паралельно виконує класифікацію об'єктів, регресію обмежувальних рамок і генерацію піксельних масок сегментації для кожного об'єкта. На рисунку 2.6 наведено схему архітектури такої моделі.

Навчання моделі засноване на комбінованій функції втрат, що враховує всі три завдання. Особливість Mask R-CNN полягає у високій якості сегментації на рівні пікселів і стійкості до складних умов зображення, що робить її затребуваною для застосунків, які потребують точного виділення об'єктів, як-от медична візуалізація або промисловий контроль якості. Головним недоліком Mask R-CNN є вища обчислювальна складність і повільний час інференсу, що обмежує використання моделі в задачах із жорсткими вимогами щодо часу.

Таким чином, розглянуті моделі реалізовують різні підходи до сегментації зображень, мають свої переваги та недоліки. YOLOv11 та її найближчі версії забезпечують вищу швидкість обробки і порівнянну або кращу точність порівняно з Mask R-CNN, що робить її кращим вибором у застосунках, де важлива оперативність. Водночас Mask R-CNN зберігає лідерство в завданнях, що вимагають максимально точної і детальної сегментації об'єктів, незважаючи на велике обчислювальне навантаження. Вибір моделі має базуватися на вимогах конкретного завдання і доступних обчислювальних ресурсах.

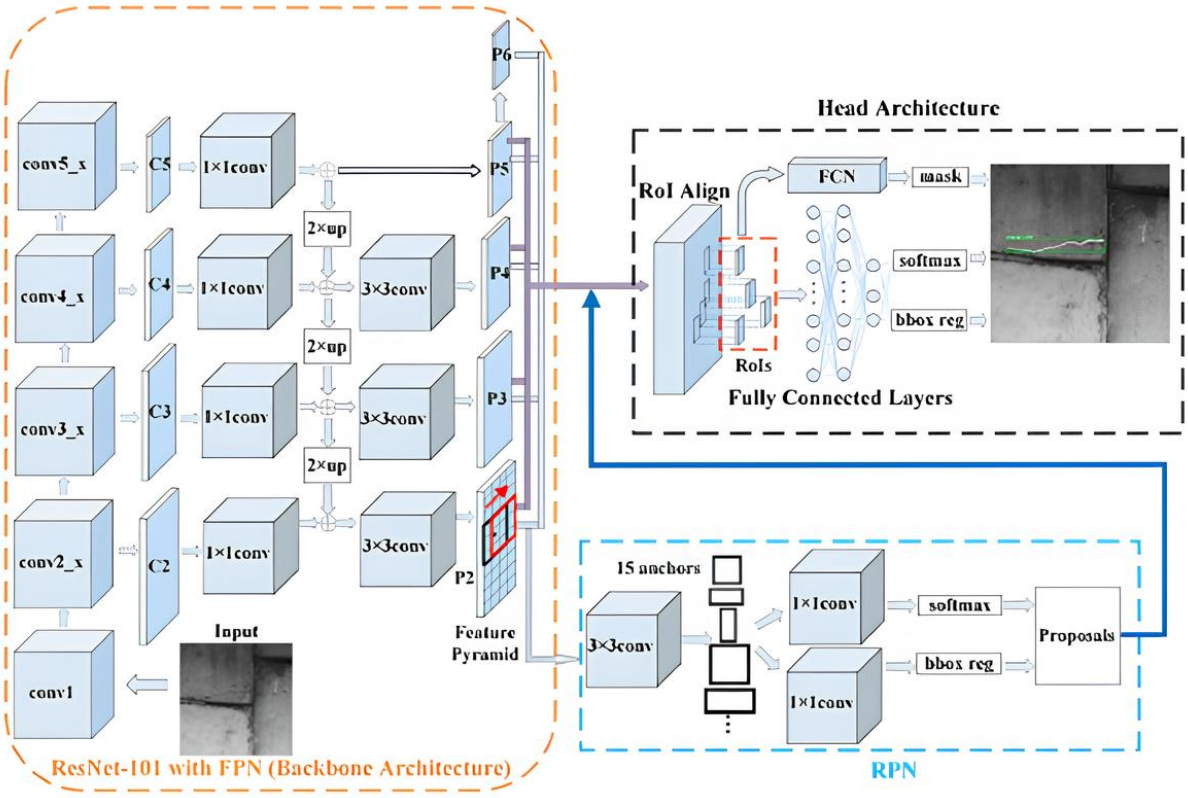


Рисунок 2.6 – Архітектура Mask R-CNN

2.5 Шляхи вирішення проблеми дисбалансу класів

У процесі створення системи автоматичного виявлення та пошкоджень будівель в ході воєнних дій в України виникає суттєва

проблема, що пов’язана з нерівномірним розподілом класів у тренувальному наборі даних. Дисбаланс класів зумовлений природною неоднорідністю типів пошкоджень, що спостерігаються на зображеннях. Зокрема, деякі категорії пошкоджень, такі як розбиті вікна, є значно більш поширеними через їхню високу частоту появи. Наприклад, навіть у разі відсутності прямого влучення в будівлю, вибухова хвиля часто призводить до пошкодження вікон, що суттєво збільшує частку таких об’єктів у наборі даних. Водночас будівлі з критичними руйнуваннями, такими як обвалені конструкції, зазвичай також мають розбиті вікна, що додатково посилює переважання цього класу. Оскільки вікна є численними в кожній будівлі, кількість прикладів цього класу значно перевищує інші категорії пошкоджень, такі як тріщини чи обвалені стіни.

Такий нерівномірний розподіл даних створює передумови для зміщення моделі в бік домінуючого класу. У разі використання стандартних підходів до навчання, модель буде демонструвати високу загальну точність, оскільки більшість анотацій у наборі даних належать до переважаючого класу, і їх правильна класифікація забезпечує високий показник точності. Проте, при використанні моделі на практиці, ефективність моделі буде суттєвого знижена, оскільки така модель буде нездатна виявити менш представлені, але критично важливі класи, які зазвичай мають вирішальне значення для оцінки ступеня руйнування та планування реконструкції. Таким чином, у даному випадку висока загальна точність не відображає реальної здатності моделі вирішувати поставлену задачу, що знижує її практичну цінність.

Для вирішення проблеми дисбалансу класів можна застосовувати як підходи для попередньої роботи з набором даних, так для налаштування методів комп’ютерного зору. Так, одним із базових та ефективних методів боротьби з дисбалансом класів шляхом попереднього налаштування набору зображень у задачах комп’ютерного зору є аугментація даних. Такий підхід дає змогу не тільки збільшити загальний обсяг навчальної вибірки, а й

збалансувати представленість мінорних класів за рахунок варіювання структури та зовнішнього вигляду зображень.

У науковій роботі [42], яка пропонує повний і детальний огляд підходів до аугментації зображень, вказується, що всі методи аугментації можна умовно розділити на базові та просунуті. Базовий підхід включає в себе методи маніпуляції зображеннями, а саме геометричні та негеометричні підходи, і методи стирання зображень.

Геометричні трансформації, основними способами яких є обертання, зрушення, масштабування, зрушення по осі, зрушення кута, змінюють просторове положення об'єктів, даючи змогу підвищити інваріантність моделі до орієнтації та положення об'єктів. З одного боку, такі трансформації дають змогу підвищити стійкість моделі з огляду на те, що в реальності пошкодження можуть бути зафіксовані під різними кутами зйомки або в різних положеннях. З іншого боку, трансформації, які кардинально змінюють просторове розташування об'єктів у кадрі, можуть призвести до створення тренувальних зображень, які не зустрічаються в реальності та можуть збити модель, що призведе не просто до відсутності позитивної зміни точності, а до значного погіршення якості. Загалом вибір і використання кожного способу геометричної трансформації вимагає всебічної попередньої перевірки на предмет відповідності отриманих у результаті його використання зображень реальності.

До негеометричних методів трансформації належать зміна колірного простору, яскравості, контрасту, ін'єкція шуму та фільтрація, що дає змогу поліпшити стійкість моделі до варіацій освітлення та текстурних шумів. На відміну від геометричних трансформацій такий підхід є більш безпечним для використання і як правило не має такого сильного впливу на зниження точності передбачення моделі. Використання такого підходу аугментації даних дозволяє зробити модель стійкішою до випадків, коли зображення, були отримані за різних умов знімання та на різні пристрої з різною якістю одержуваних знімків. При цьому важливим є не зловживати з

використанням надмірної кількості таких методів, що може призвести до значного погіршення якості зображень і нездатності моделі розрізнити потрібні класи під час навчання.

Методи стирання, такі як CutOut, Random Erasing і GridMask, передбачають навмисне видалення випадкових або регулярних ділянок зображення, що змушує модель приділяти більше уваги різним частинам об'єкта, а не тільки найбільш помітним ознакам, і тим самим сприяє кращому розпізнаванню рідкісних або малопомітних об'єктів.

До просунутих технік аугментації зображень відносять методи змішування зображень, такі як Mixup, CutMix, SaliencyMix і PuzzleMix, які створюють синтетичні зображення, об'єднуючи ділянки з двох і більше зображень, при цьому пропорційно об'єднуючи і мітки, і методи на основі самонавчання, такі, як AutoAugment, Fast AutoAugment, що здатні автоматично підбирати оптимальні стратегії аугментації.

На всіх зображеннях представлені будівлі з ушкодженнями, при цьому при сильному пошкодженні будівлі під час воєнних дій – руйнування частини будівлі є присутніми і дрібні руйнування такі, як розбиті або знищені віконні рами. Виходить, що мінорні класи (зруйновані конструкції, тріщини, уламки штукатурки) і мажорні класи (розбиті вікна) є сусідами в одних і тих самих кадрах, і просте множення кількості вихідних зображень лише пропорційно збільшує частку і тих, і інших об'єктів, не змінюючи їхнього відносного балансу.

В умовах обмеженого інструментарію, наприклад під час використання базового плану Roboflow, вирішити цю проблему можна за рахунок регіонально-виборчих трансформацій [43]. По-перше, цільовий кропінг (targeted cropping) дає змогу створити з вихідних знімків окремі фрагменти, сконцентровані на мінорних класах: вирізані ділянки з тріщинами або уламками стають самостійними прикладами, що підвищує частоту їхньої появи в навчальній вибірці та посилює внесок відповідних градієнтів під час оптимізації. Враховуючи зазначену проблему з наявністю

на одному зображенні мінорних та мажорних класів, а також певну обмеженість підходів аугментації на Robobflow, має сенс розглянути іншими підходи вирішення проблеми с дисбалансом класів.

Для вирішення проблеми дисбалансу класів можливе застосування модифікованих функцій втрат, спрямованих на посилення внеску помилок, що виникають під час розпізнавання рідкісних класів. В статті [44] розглядаються різні функції втрат, які ефективні саме при сегментації екземплярів. Автори наукової статті звертають увагу на використання зваженої бінарної крос-ентропії, фокальної функції втрат, а також більш просунутих функцій втрат таких, як Generalized IoU та Boundary Loss.

Складнішим, але ефективнішим підходом у боротьбі з дисбалансом класів є підхід двофазного навчання (Two-phase Learning), згідно з яким модель спочатку навчається на збалансованій підвибірці, а потім донавчається на повному, незбалансованому датасеті [45].

На першому етапі за допомогою методів, таких як Random Under Sampling і аугментація мінорних класів, створюється більш збалансована вибірка, що дає змогу моделі ефективно навчатися розпізнавати рідкісні об'єкти. Після цього модель донавчається на повному наборі даних із природним розподілом класів, щоб скоригувати передбачення з урахуванням реальної статистики та поліпшити узагальнюючу здатність.

Перевагою такого підходу є можливість виділення уваги рідкісним класам без втрати інформації про реальний розподіл. Метод є універсальним і легко поєднується з іншими техніками боротьби з дисбалансом, такими як спеціалізовані функції втрат. Основними недоліками такого підходу є його зростаюча тривалість і ресурсомістність, що пояснюється навчанням моделі двічі на різних підвибірках. Загалом, незважаючи на таке обмеження, двофазне навчання залишається ефективним і широко застосовуваним підходом для підвищення якості моделей у задачах із вираженим дисбалансом класів.

2.6 Методи ранжування

Визначення пріоритетності відновлення пошкоджених об’єктів міської інфраструктури потребує використання методів ранжування, які забезпечують об’єктивне упорядкування споруд за різними множинними критеріями. У сучасних дослідженнях для вирішення таких завдань застосовуються методи MCDM та алгоритми LTR, які дозволяють інтегрувати різнорідні дані та оптимізувати процес прийняття рішень. Обидва підходи знаходять застосування у практичних задачах планування, але мають відмінності у вхідних даних, складності реалізації, можливостях масштабування та рівні прозорості результату. Для їх аналізу розглянемо типових представників кожної групи алгоритмів.

MCDM-методи широко застосовуються у сфері управління, планування міської інфраструктури, екології та стратегічного аналізу. Їх основною перевагою є можливість формалізованого врахування різнорідних оцінок (кількісних і якісних), а також зручність реалізації без потреби в попередньому навчанні моделі [46].

Одним з найпоширеніших методів зазначеної групи алгоритмів є алгоритм TOPSIS [47]. Зазначений метод передбачає вибір альтернативи, яка знаходиться найближче до умовно найкращого рішення та одночасно надалі від умовно найгіршого. В якості вхідних даних розглядаються наступне: нехай задано m альтернатив та n критеріїв, тоді на першому етапі необхідно представити вхідні дані у вигляді матриці рішень, яка містить значення кожного критерія j для кожної альтернативи i.

$$X = [x_{ij}], \tag{2.1}$$

де $i = 1, \dots, m$ – i – альтернатива,

$j = 1, \dots, n$ – j – критерій.

На наступному кроці алгоритму відбувається формування матриці нормалізованих значень $R = [r_{ij}]$ шляхом нормалізація побудованої на попередньому кроці матриці рішень з метою усунення впливу різних одиниць виміру на визначення пріоритетів.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}. \quad (2.2)$$

Для урахування відносної важливості кожного з критеріїв до елементів нормалізованої матриці, отриманої на попередньому кроці застосовуються ваги критеріїв w_j. На даному кроці відбувається формування зваженої нормалізованої матриці:

$$V = [v_{ij}], \quad (2.3)$$

де $v_{ij} = w_j \times r_{ij}$;

w_j – вагове значення j-ї критерію;

r_{ij} – нормалізоване значення j-го критерію для і-ї альтернативи.

Далі відбувається визначення найкращого та найгіршого рішення шляхом визначення для кожного критерію відповідного значення.

$$A^+ = \{v_1^+, \dots, v_n^+\}, \quad (2.4)$$

$$A^- = \{v_1^-, \dots, v_n^-\}, \quad (2.5)$$

де $v_j^+ = \max\limits_i v_{ij}$ – найкраще значення критерію;

$v_j^- = \min\limits_i v_{ij}$ – найгірше значення критерію.

На цьому етапі закінчуються підготовчі кроки та починається сам процес ранжування. Для кожної альтернативи розраховуються евклідові відстані до попередньо визначених ідеальної та антиідеальної точок.

$$D_i^+ = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^+)^2}, \tag{2.6}$$

$$D_i^- = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^-)^2}, \tag{2.7}$$

де D_i^+ – відстань від альтернативи і до ідеального рішення;

D_i^- – відстань від альтернативи і до антиідеального рішення.

Закінчується виконання алгоритму розрахунком для кожної альтернативи коефіцієнта відносної близькості до ідеального рішення C_i.

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-}, \tag{2.8}$$

$$0 \leq C_i \leq 1. \tag{2.9}$$

При цьому вважається, що чим більше значення отримане значення, тим ближча альтернатива до ідеалу. Отже найкраща альтернатива та, що має максимальне значення коефіцієнта відносної близькості.

Результатом роботи даного алгоритму є список пріоритетів що містить альтернативи впорядковані за спаданням коефіцієнта відносної близькості.

Іншим широковживаним MCDM-методом є алгоритм AHP, який був запропонований Томасом Сааті у 1970-тих роках, і застосовується для структурування складних задач і ранжування альтернатив на основі множинних критеріїв. Метод AHP базується на декомпозиції задачі прийняття рішень на ієрархічну структуру, що складається з мети, критеріїв,

субкритеріїв та альтернатив. В статті [48] детально описується принцип роботи алгоритму, який реалізується послідовно в кілька етапів. На першому кроці на основі результатів експертних опитувань формується матриця парних порівнянь A = $[a_{ij}]$, в якій кожен елемент відображає відносну значущість критерію i у порівнянні з критерієм j. При цьому при побудові такої матриці враховуються принципи взаємності та порівняння критерію самого з собою. На наступному кроці відбувається формування вектору нормованих ваг, як головного власного вектору матриці A.

$$Aw = \lambda_{max} w, \tag{2.10}$$

$$\lambda_{max} \geq n, \tag{2.11}$$

де λ_{max} – це найбільше власне значення матриці A;

w – нормалізований головний власний вектор

n – кількість критеріїв.

Для визначення вектору нормованих ваг важливим є попереднє обчислення найбільшого власного значення λ_{max}.

$$\lambda_{max} = \frac{\sum a_j w_j - n}{w_l} \tag{2.12}$$

Для обчислення вектору w також можливим є використання методу степеневих множень.

$$e = (1, 1, \dots, 1)^T, \tag{2.13}$$

$$w = \lim_{k \to \infty} \frac{A^k e}{e^T A^k e}. \tag{2.14}$$

Для перевірки результатів роботи алгоритму та узгодженість парних порівнянь відбувається розрахунок індексу узгодженості (CI) та співвідношення узгодженості (CR).

$$CI = \frac{\lambda_{max} - n}{n-1}, \tag{2.15}$$

$$CR = \frac{CI}{RI}, \tag{2.16}$$

де RI – середній індекс узгодженості випадкової матриці розміром n × n.

Результатом роботи методу AHP стає вектор вагових коефіцієнтів критеріїв, який відображає відносну важливість кожного критерію у прийнятті рішення, значення глобальних пріоритетів альтернатив та упорядкований список альтернатив в порядку від найбільшої до найменшої пріоритетної, відповідно до обчислених глобальних пріоритетів.

Як зазначається в науковій роботі [49], даний метод має як свої переваги, так і недоліки. До головних його переваг відносяться гнучкість, що походить з можливості методу AHP легко адаптується до різних задач багатокритеріального аналізу, враховуючи як кількісні, так і якісні критерії; підтримка колективної участі експертів у прийнятті рішень шляхом обчислення середньої геометричної індивідуальних парних порівнянь; стійкість до невизначеності та ризику, що зумовлена здатністю формування шкали оцінок у випадках, коли стандартних вимірювань недостатньо, присутні неповні або розрізнені данні; зрозумілість процесу розрахунку рангу та отриманих результатів.

Однак, згідно з вже згаданої статтею, даний алгоритм є довгим та трудомістким для проведення через необхідність виконання численних парних порівнянь, особливо у випадках з великою кількістю критеріїв чи альтернатив. Іншим суттєвим недоліком є проблема порушення пріоритетності об’єктів при додаванні до множини альтернатив дуже

схожих або копій існуючих варіантів. Через адитивний спосіб агрегування оцінок високі показники за одними критеріями можуть компенсувати низькі за іншими, що може призвести до надмірного згладжування результатів та втрати важливих деталей. Деякі фахівці також вважаються за недолік обмеженість інтенсивності пріоритету дев'ятибальною шкалою, яка в деяких випадках може виявитись недостатньою.

У випадках, коли доступна історична інформація про ранжування об'єктів або існує експертна оцінка, є доцільним використання підходів LTR. Ці методи широко застосовуються в інформаційному пошуку, рекомендативних системах, сортуванні заявок і можуть бути адаптовані до задач міського управління [50].

Алгоритми RankNet, LambdaRank і LambdaMART є важливими представниками такого напрямку, активно використовуються для побудови моделей, що сортують або ранжують документи чи альтернативи відповідно до їхньої релевантності або значущості [51]. Ці методи еволюціонували від простого порівняння пар документів до більш складних підходів, що напряму оптимізують метрики списку, такі як NDCG. Важливим аспектом розвитку цих алгоритмів є їх здатність оптимізувати ранжування через градієнтний спуск за допомогою різних функцій втрат, що враховують специфіку ранжування. Алгоритм RankNet, запропонований компанією Microsoft Research, є першим методом у лінійці LTR і ґрунтується на підході pairwise. Основна ідея полягає в тому, що модель порівнює пари документів і визначає, який із них є більш релевантним. Для кожної пари документів (d_i, d_j), де d_i має бути вище в ранжуванні за d_j, RankNet використовує сигмоїдну функцію для прогнозування ймовірності того, що d_i буде кращим за d_j:

$$P_{ij} = \frac{1}{1+e^{-\sigma(s_i - s_j)}}, \tag{2.17}$$

де s_i та s_j – це скорингові значення, що генерує модель для документів d_i та d_j відповідно,

σ – параметр моделі.

Функція втрат базується на крос-ентропії, що мінімізується для того, щоб навчити модель робити коректні прогнози. Однак, цей підхід орієнтований на мінімізацію помилки за кожну пару документів, а не безпосередньо за метрикою ранжування, що є основним обмеженням цього методу.

LambdaRank є покращенням RankNet і вирішує важливу проблему – він впроваджує лямбда-градієнти, які дозволяють оптимізувати не просто точність за парами, а саме метрики якості списку, такі як NDCG. Ідея LambdaRank полягає в тому, що, замість того щоб коригувати модель лише на основі помилок в парних порівняннях, градієнти коригуються таким чином, щоб мінімізувати зміни в метриці якісного списку, що спричиняють перестановки пар документів. Це дозволяє напряму оптимізувати ранжування, зменшуючи ймовірність того, що релевантні документи опиняться в нижніх позиціях списку. Однак LambdaRank залишається нейронною мережею, що робить його менш ефективним при роботі з табличними даними та зменшує прозорість результатів.

LambdaMART є подальшим розвитком LambdaRank і поєднує ідеї λ- градієнтів з градієнтним бустингом на MART-деревах. Цей алгоритм використовує ансамбль дерев рішень, де кожне дерево коригує помилки попереднього. Кожне дерево навчається на залишках помилок, що були допущені в попередніх дереві, і таким чином оптимізує метрику ранжування, враховуючи значення λ-градієнтів. Завдяки цьому LambdaMART напряму оптимізує метрики списку, такі як NDCG, та дає точніші результати при великому обсязі даних. Він ефективно працює з як категоріальними, так і числовими даними, що робить його дуже гнучким і масштабованим. Однак, його складність полягає в необхідності глибокого налаштування параметрів і потенційному перенавчанні без застосування регуляризації.

Алгоритми RankNet, LambdaRank і LambdaMART послідовно вдосконалюють підхід до Learning to Rank, від простих моделей, що працюють з парами документів, до складніших методів, які можуть оптимізувати якість списку за допомогою бустингових методів. LambdaMART, у порівнянні з попередніми методами, є найбільш потужним, оскільки він об’єднує силу λ-градієнтів і градієнтного бустингу на деревах, що дозволяє значно покращити точність ранжування при великих наборах даних і з високою варіативністю. Однак усі ці методи потребують великих обчислювальних ресурсів і налаштувань для уникнення перенавчання.

Для пріоритизації відновлення будівель методи MCDM (TOPSIS, AHP) є хорошим вибором, якщо є кілька критеріїв, але немає розмічених даних, оскільки вони забезпечують простоту та прозорість. RankNet, LambdaRank і LambdaMART підходять для великих наборів даних з попередніми оцінками, оскільки вони орієнтовані на оптимізацію метрик ранжування, таких як NDCG. Для задачі з обмеженими даними або експертними оцінками методи MCDM будуть зручнішими. Метод TOPSIS є простим у реалізації, забезпечує чітке і інтуїтивно зрозуміле ранжування альтернатив на основі їх відстані до ідеального і антиідеального рішень. Він також добре працює, коли є обмежена кількість даних і не потрібно глибокої моделювання залежностей між критеріями. TOPSIS дає змогу легко інтерпретувати результати і приймати обґрунтовані рішення на основі оцінок за кількома критеріями.

3 ПРОЕКТУВАННЯ СИСТЕМИ ТА ЇЇ РОЗРОБКА

3.1 Проєктування системи

На рисунку 3.1 представлено схему системи визначення черговості відновлення будівель під час війни та після її завершення.

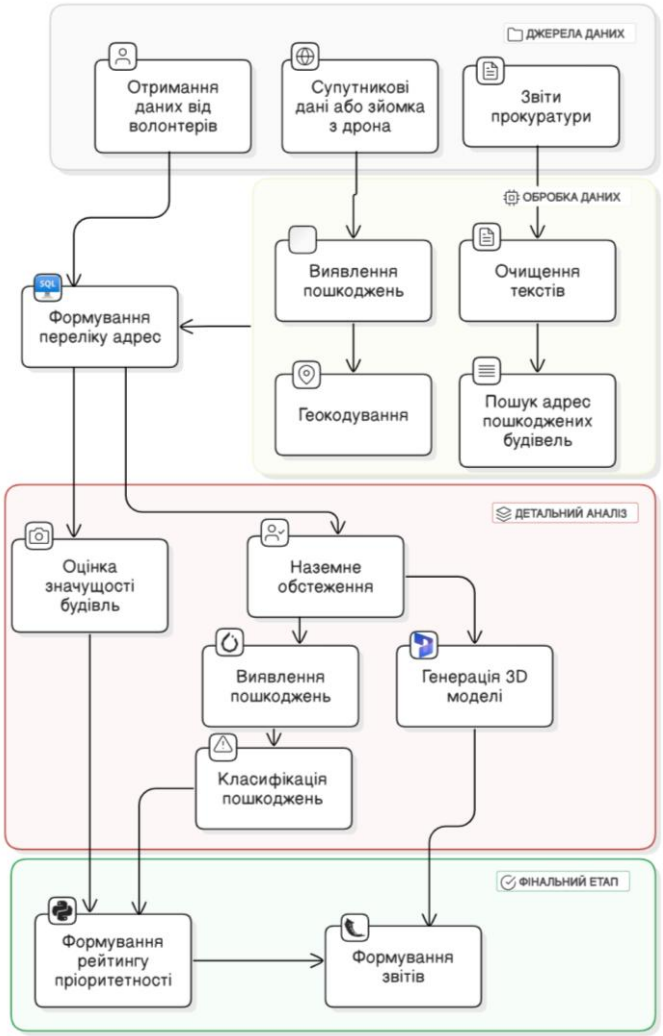


Рисунок 3.1 – Загальна схема системи визначення черговості відновлення будівель

Як можна побачити на рисунку, процес визначення черговості відновлення будівель складається з 3 великих етапів: спершу здійснюється

робота з джерелами даних для виявлення всіх пошкоджених будівель та формування їхнього переліку адрес, далі проводиться збір та аналіз детальної інформації по кожній споруді зі списку та фінальному етапі відбувається створення списку всіх обстежених будівель відсортованих за пріоритетністю їхнього відновлення та формування звіту, що включає результати роботи системи. Ідея такої системи описується у власних тезах [52].

Для отримання інформації про всі пошкоджені будівлі та стійкості системи до можливих проблем з якимось певним джерелом даних, в системі передбачається робота з 3 різними незалежними один від одного первинними джерелами.

Для швидкого аналізу великих територій в системі передбачається робота з супутниками та вирішення задачі класифікації об’єктів: розпізнавання будівель на зроблених знімках місцевості та присвоювання їм класу, що позначає рівень пошкодження. Після завершення роботи методу комп’ютерного зору в системі починається наступний крок пов'язаний з прив’язкою знайденою будівлі до її адреси та формування списку адрес з позначками рівня руйнування будівлі. Враховуючи те, що при такій зйомці будівлі, як правило, обстежуються лише зверху, то існує можливість пропуску будівель, що мають цілий дах, але руйнування стін та фасаду. Для урахування зазначеного типу будівель в системі передбачена робота з двома іншими джерелами даних.

Як правило, на місцях уражень відразу починають працювати операційні служби, які фіксують пошкодження будівель. При співпраці з державними органами зі звітів таких служб, зокрема прокуратури, можна отримати адреси, фото пошкоджень знятих з різних ракурсів, або, навіть, певну оцінку ступеню пошкодженості споруди. Використання таких звітів в системі дозволяє значно уточнити список зруйнованих будівель, отриманих при аналізі супутникових знімків. Однак для деокупованих територій, що мають значні за площею та серйозністю руйнувань, однак достатня кількість

зазначених звітів для них відсутня. В такому випадку важливу роль відіграє останній інформаційний ресурс – співпраця з волонтерами та небайдужими громадянами, які здатні повідомити через налагоджені канали зв'язку повідомити про знаходження ураженої будівлі за певною адресою, суб'єктивно оцінити серйозність пошкоджень або, навіть, надіслати фотозвіти. Отже, результатом роботи першого блоку системи є сформований після аналізу супутникових знімків перелік адрес, відсортований за позначками ступеню руйнування, доповнений інформацією, отриманою від операційних служб та волонтерів.

Після формування зазначеного списку система переходить до виконання другого етапу. На цьому етапі відбувається пошук та збір всієї доступної інформації для кожної з будівель у сформованому списку. Даний етап можна розділити на два блоки, що не залежать один від одного, а тому можуть виконуватись паралельно. Перший блок полягає в роботі з архівами та базами даних для отримання повної інформації про історію зведення будівлі, її архітектурний стиль, архітектора, важливі історичні події, що можуть бути з нею пов'язаними, інформацію про важливість даної будівлі для соціуму та її функціональна роль в даний момент. Останнім кроком є пошук даних про тип споруди. Більшість будівель будуються за типовими проектами та зазвичай будівельна інформація, що характерна для типового типу та серії будинків, є правдивою і для самої споруди. Далі за певними правилами відбувається визначення соціально-культурної цінності та соціального значення будівлі. Обидві оцінки представляються за шкалою від 1 до 10.

Другий блок такого етапу полягає в споряджені мобільних груп, що виїжджають за всіма адресами та проводять наземне обстеження, під час якого проводять зйомку будівлі зі всіх ракурсів для отримання найбільш повної інформації про її стан. Допустимим на даному кроці є й проведення запису відео повного обходу будівлі, якщо таке є можливим. Після внесення отриманої інформації в систему починається етап візуального

автоматичного аналізу за допомогою методів комп’ютерного зору для виявлення різних типів пошкоджень. На наступному кроці відбувається визначення ступеню пошкодження будівлі шляхом співвідношення визначених різних типів руйнувань з десятибальною шкалою, від незначних до критичних ушкоджень.

Отримані під час обстеження відеоматеріали обробляються окремо за допомогою алгоритмів із реалізацією Structure-from-Motion (SfM) реконструкції, в результаті чого створюється детальна 3D-модель споруди, що дозволяє фахівцям більш детально попередньо оцінити обсяг ремонтних робіт без необхідності фізичного виїзду на об’єкт.

На фінальному етапі реалізований в системі метод ранжування інтегрує отриманні раніше показники (рівень руйнувань і культурна історична-цінність та соціальна значущість) у єдину метрику пріоритетності.

Результатом роботи всієї системи є відсортований за пріоритетом список пошкоджених об’єктів та детальні звіти по кожному з них, що містять інформацію про типи дефектів, 3D-модель і оцінку соціокультурної цінності.

3.2 Структура проєкту

Описану в попередньому пункті систему визначення пріоритетності відновлення будівель з урахуванням автоматично виявлених пошкоджень, розрахованого балу серйозності пошкодження, її соціального значення та культурної цінності реалізовано у вигляді клієнт-серверного веб-застосунку, серверну частину якого реалізовано на Flask. При побудові цього проєкту підтримувався принцип розподілення обов’язків, тобто структура проєкту організована таким чином, щоб забезпечити його модульність, зручність обслуговування та можливість розширення

функціональності в майбутньому. На рисунку 3.2 представлено вигляд структури проекту в середовищі розробки.

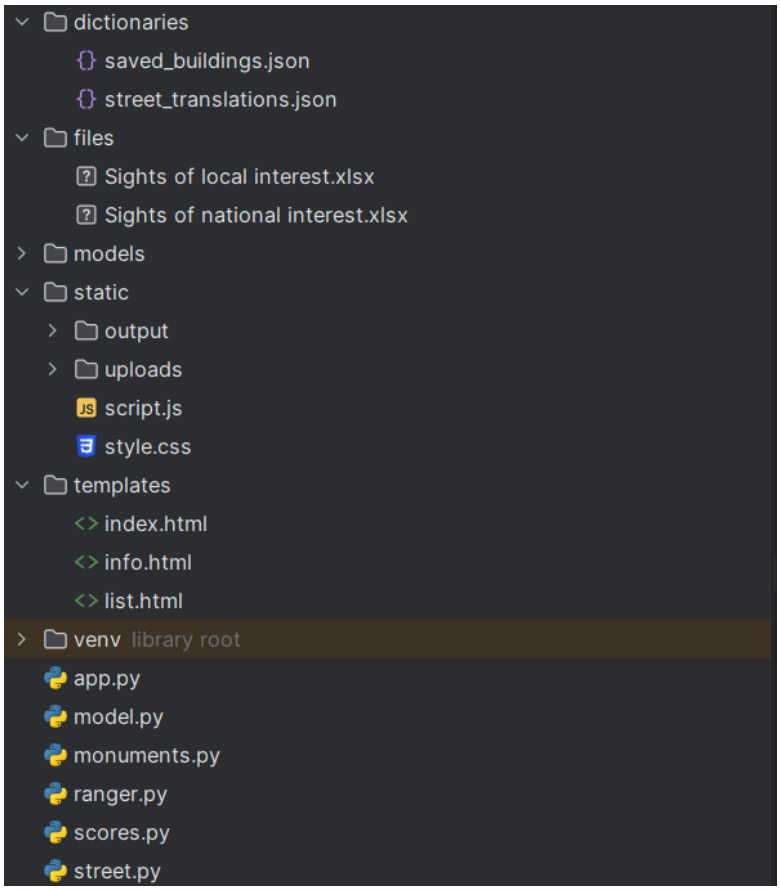


Рисунок 3.2 – Структура проєкту

Проєкт містить наступні складові частини:

– директорія dictionaries містить файли з довідковою інформацією, необхідною для обробки даних. Файл saved_buildings.json зберігає записи з необхідною для проведення ранжування інформацією про пошкоджені будівлі. Такий файл слугує базою даних, читання та запис в нього відбувається тоді, коли виникає потреба в виведенні переліку всіх пошкоджених адрес на веб-сторінку та додання запису після проведення інспектування нового пошкодженого будинку. Файл street_translations.json є користувацьким словником та містить співвідношення між старими та новими назвами вулиць міста Харкова, також підтримуючи можливість

правильного перекладу назв вулиць з різних мов на українську. Даний файл є важливим в застосунку для отримання всієї потрібної інформації про будівлю, яка надходить з різних джерелах, де зберігається в різних форматах та має різні позначення назв вулиць;

– у директорії files знаходяться xlsx-файли, що були завантажені з офіційної сторінки Міністерства культури та стратегічних комунікацій України і є частиною державного реєстру нерухомих пам’яток України. В зазначеній директорії зберігаються файли Sights_of_local_interest.xlsx та Sights_of_national_interest.xlsx, які містять інформацію про об’єкти місцевої та національної культурної цінності відповідно. Ці файли слугують джерелом даних для оцінки культурно-історичної значущості будівель;

– директорія models містить ваги попередньо натренованих моделі глибокого навчання, які можуть бути використаними для виявлення пошкоджень на фотографіях будівель. Ця директорія є критичною для автоматизації процесу аналізу зображень та формування словника, що відображає класи пошкоджень та кількість виявлених екземплярів цього класу на ображенні;

– директорія static зберігає статичні ресурси веб-додатку. Вона містить піддиректорії output та uploads, які використовуються для зберігання результатів обробки зображень моделлю комп’ютерного зору та завантажених користувачем фотографій відповідно. Окрім того, файли script.js та style.css реалізують динамічну поведінку та стилізацію клієнтської частини інтерфейсу застосунку;

– директорія templates містить HTML-шаблони для рендерингу веб-інтерфейсу. Такі файли, як index.html, info.html та list.html забезпечують відображення головної сторінки, детальної інформації про будівлі та списку результатів відповідно;

– директорія venv library root представляє віртуальне середовище Python, яке використовується для ізоляції залежностей проєкту.

В головній директорії проекту знаходяться скрипти, які й є основною частиною проєкту. Для того, щоб забезпечити гарний тон при створені проєкту всі функціональності проєкту були представлені в різних python- файлах. Такими файлами в системі є:

– app.py, що є головним файлом застосунку, який ініціалізує Flask- сервер, містить прописані маршрути для пов’язування роботи серверу з клієнтською частиною та загальну логіку роботи веб-інтерфейсу;

– model.py є модулем, відповідальний за завантаження та використання попередньо натренованої моделі глибокого навчання для виявлення пошкоджень на зображеннях;

– monuments.py містить логіку для роботи з даними про історико- культурні об’єкти, а також обробку відповідних Excel-файлів;

– ranger.py реалізує механізм багатокритеріального ранжування об’єктів за допомогою методу TOPSIS;

– scores.py є модулем для обчислення трьох ключових оцінок будівлі: серйозності пошкоджень, функціонального значення та культурної цінності;

– street.py містить функції для обробки адрес, включаючи переклади назв вулиць, пошук по словниках, виконання звернень для різних баз даних та архівів для отримання всієї необхідної інформації про будівлю.

3.3 Підготовка набору даних

Результат роботи моделі комп’ютерного зору, що використовується для визначення пошкоджень на зображеннях будівель, значною мірою залежить від якості тренувального набору даних. Отже, постає завдання в підготовці такого набору даних з достатньою кількістю зображень, що будуть правильно проанотовані. Враховуючи, що існує невелика кількість зображень високої якості було прийнято рішення поєднати декілька відповідних наборів даних [53], [54]. Однак виникла проблема з якістю та

детальністю анотації в таких наборах зображень, що було неприпустимим для поставленої задачі. В обраному наборі даних для позначення руйнування будівель було визначено загальний клас damage, який використовувався для будь-яких пошкоджень не враховуючи їх серйозність. Для того, щоб дозволити моделі бути більш гнучкою у виявленні різних типів пошкоджень, а також враховуючи той факт, що визначення ступеню пошкодження будівель винесено за рамки моделі комп'ютерного зору, було прийнято рішення про самостійну анотацію зображень.

Робота з набором даних відбувалась на платформі Roboflow [55] з використанням напіватоматизованих інструментів розмітки. На рисунку 3.3 наведено вигляд інтерфейсу середи для анотування зображень, де справа знаходиться саме розмічене зображення, а зліва знаходиться інформації про мітки визначених класів.

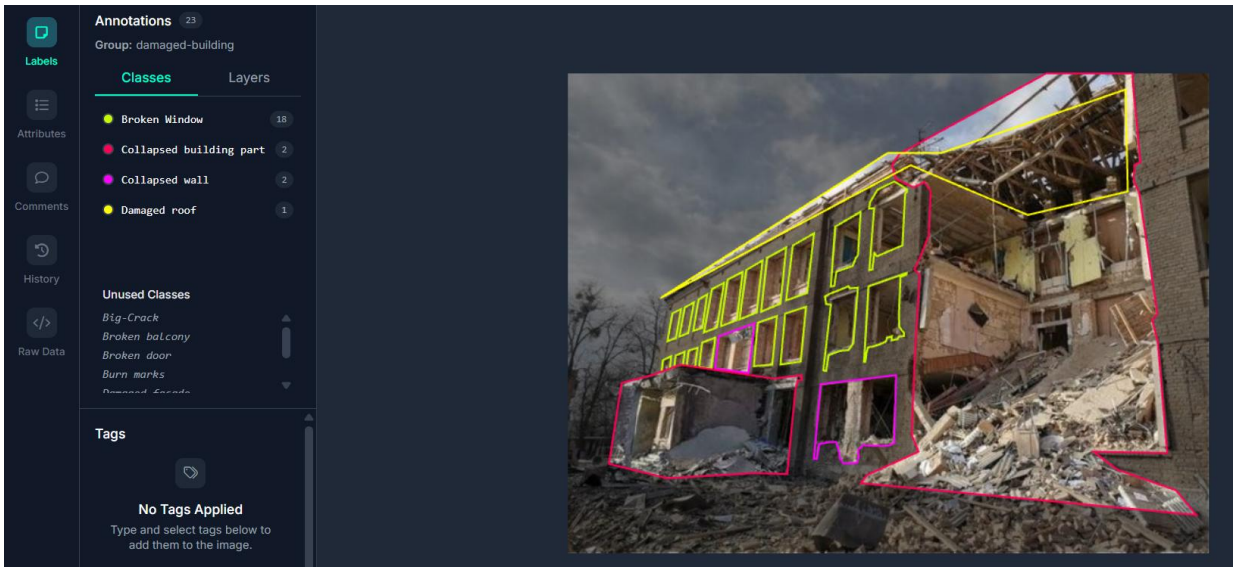


Рисунок 3.3 – Вигляд інтерфейсу середи розмітки на Roboflow

В ході розмітки зображень постала серйозна проблема у розробці повного переліку пошкоджень для найбільш точної оцінки визначення різних класів руйнувань, що відрізняються різним ступенем тяжкості. Після проведення детального аналізу різноманітних звітів обстеження

пошкоджених будівель, а також візуального огляду наявного набору даних, було вирішено визначити такі класи: велика тріщина (Big-Crack), середня тріщина (Medium-Crack), мала тріщина (Small-Crack), зруйнована стіна (Collapsed wall), зруйнована частина будівлі (Collapsed building part), зруйнований дах (Damaged roof), згорілий фасад (Burn marks), пошкоджений фасад (Damaged facade), пошкоджений балкон (Broken balcony), зруйновані двері (Broken door), зруйноване вікно (Broken window).

Визначений перелік зазначених класів дозволяє охопити велику кількість пошкоджень, при цьому не має виникнути проблема з суб'єктивністю визначення приналежності пошкодження до того чи іншого класу. Для остаточного уникнення плутанини при розмітці набору даних було прийнято рішення рахувати пошкодженим фасадом порушення верхнього шару будівлі, наприклад, облицівки, зруйнованою стіною – порушення цілісності стіни, а зруйнованої частиною будівлі – порушення двох або більше стін, що знаходяться поруч. Згорілий фасад є різновидом пошкодженого фасаду, визначеним для відокремлення пошкодження саме вогнем, що не має безпосереднього впливу на стійкість споруди, але може свідчити про наявність внутрішніх порушень. На рисунку 3.4 для порівняння зображено фото з початковою розміткою (зліва) та ті, що вийшли після завершення процесу власної анотації.

Як можна побачити, в наборі даних, який використовувався далі в даній кваліфікаційній роботі для тренування моделей комп'ютерного зору представлено більше різноманіття класів пошкоджень.

Після завершення розмітки набору даних в його фінальній версії залишилось 383 зображення з 11 класами анотацій. Загалом було нанесено 12973 масок анотацій. При цьому більше за всього анотацій було створено для класу розбитих вікон – 10234, інший клас з великою, але значно меншої кількістю класів – це пошкоджений фасад з 1629 анотаціями. Всі інші класи мають загальну кількість анотацій від 100 до 600. Клас з найменшою кількістю анотацій – це клас з маленькими тріщинами – 114 анотацій.

Рисунок 3.4 – Порівняння зображень з попередньою та зробленою розміткою

Як можна побачити, присутній значний дисбаланс класів, що зумовлений різною природою пошкоджень та природньою різною частотою їх появи. Такий дисбаланс може значно вплинути на точність роботи моделей для виявлення пошкоджень, змушуючи їх добре розбиратися в найбільш масових класах, та погано визначати інші класи. В подальшому, на етапі тренування моделей буде поставлено завдання вирішення зазначеної проблеми.

Після завершення анотування набору зображень доцільним стає ознайомлення з аналітичною інформацією, що автоматично генерується на платформі Roboflow. Аналіз характеристик набору даних дає змогу краще визначити, яку модель комп’ютерного зору доцільніше використовувати, а також спланувати оптимальну підготовку даних для її навчання.

На рисунку 3.5 представлено інформацію про розміри зображень. Аналізуючи дані про розмір зображення можна побачити, що більшість зображень середнього розміру, але трапляються як великі, так і маленькі зображення. Також більшість зображень є дуже широкими, що пояснюється

проведенням зйомки з дронів та спробою захватити в кадр всю будівлю. Для вирішення проблеми різної розмірності на етапі формування набору даних виникає потреба в змінені розміру зображень та уніфікації його для всіх зображень. Зазвичай зображення приводиться до розміру 640×640, але враховуючи велику кількість маленьких за розмірами пошкоджень було прийнято рішення змінити розмір зображення до 1024×1024, що сповільнить тренування моделей, але підвищить якість виявлення пошкоджень.

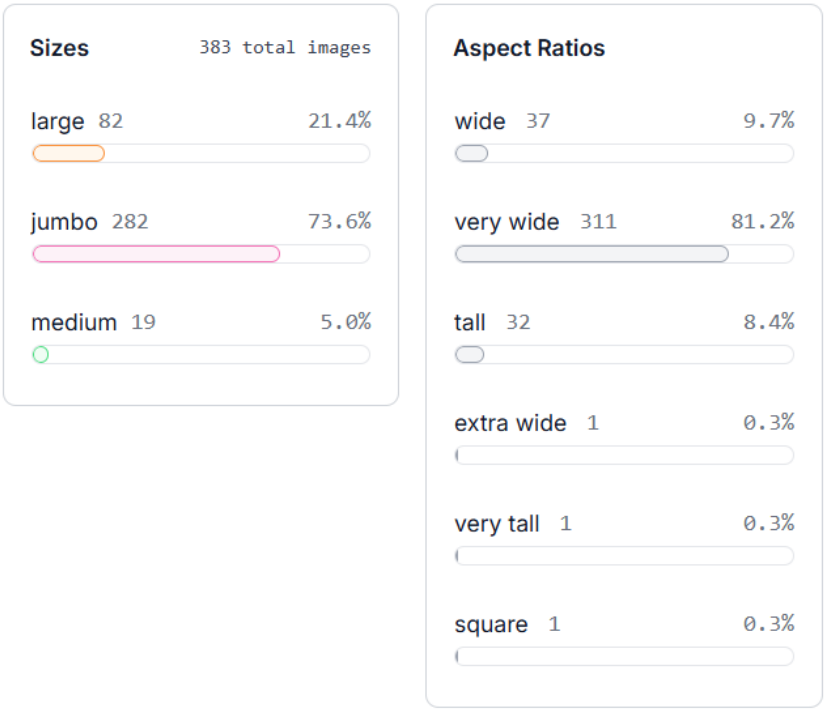


Рисунок 3.5 – Інформація про розміри та відношення ширини до висоти зображень набору

На рисунку 3.6 показано карту щільності анотацій (Annotation Heat Map), яка візуалізує розподіл анотацій по всьому простору зображень у датасеті. Як можна помітити, центральна область (червона та помаранчева зони) містить найбільшу концентрацію анотацій, тоді як периферія (синя зона) практично не містить об'єктів. Така особливість свідчить про

просторове зміщення – більшість об'єктів для виявлення знаходяться в центрі зображень. Для того, щоб уникнути перенавчання моделі на центр, коли модель очікуватиме необхідні для виявлення об'єкти в центрі зображення та потенційно не помічатиме їх в інших місцях, постає завдання в необхідності вирішення описаної проблеми.

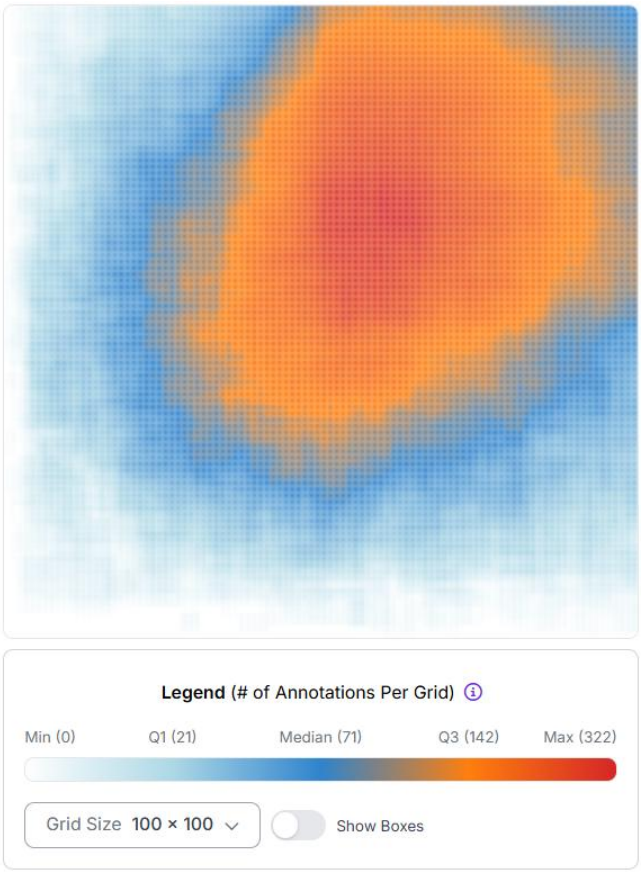


Рисунок 3.6 –Карта щільності анотацій

Для вирішення зазначеної проблеми просторового зміщення, як правило, використовують геометричні трансформації. Однак, оскільки на всіх зображеннях присутні будівлі та їхній вигляд не можна перевернути або значно змінити їхнє просторове розташування, то підходами, які може використовуватися на етапі аугментації, залишаються Crop, що буде змушувати модель бачити об'єкти в різних місцях зображення, та Shear, що дасть змогу зробити дані більш різноманітними за геометрією, але без

спотворення орієнтації. Використовуючи тільки такі процеси аугментації для розв'язання описаних раніше проблем із набором даних, було створено версію набору зображень з 696 фотознімків, на яких вже можна навчати моделі комп'ютерного зору.

Для додаткового збільшення кількості зображень можливе використання таких кроків аугментації, як зміна яскравості (brightness), розмиття зображення (blur) і додавання шуму (noise). У такий спосіб не тільки збільшується кількість зображень у наборі даних до 4048, а ще з’являється можливість зробити модель більш стійкою до мінливих умов проведення зйомки та якості зображень, які подаються їй на вхід. Отже, на даному етапі біло сформовано 2 набори зображень для навчання моделей розпізнавання пошкоджень будівель.

3.4 Модуль визначення пошкоджень

Головною функцією модулю визначення позначень є функція predict_and_visualize_return_data, яка відповідає за виконання використання ваг попередньо натренованої моделі виявлення пошкоджень до потрібних зображень та підготовку вихідних даних для подальшого аналізу і візуалізації. Програмний код даної функції наведено у лістингу 3.1

Лістинг 3.1 – Програмний код функції для використання ваг натренованої моделі Yolov11 для виявлення пошкоджень

```
def predict_and_visualize_return_data(image_path):
models = {"best_v1": YOLO("models/weights/best_v11.pt")}
original_bgr = cv2.imread(image_path)
if original_bgr is None:
raise FileNotFoundError(f"Image not found: {image_path}")
result = models["best_v1"](image_path,
device='cpu',
save=False, conf=0.5)[0]
```

Продовження лістингу 3.1

```
class_counts = Counter()total_area_pixels = 0
if result.masks is not None:
for box in result.boxes:
class_counts[result.names[int(box.cls)]] += 1
for mask in result.masks.data:
total_area_pixels += mask.cpu().numpy().sum()
annotated = result.plot(labels=False, boxes=False)
if annotated.ndim == 3 and annotated.shape[2] == 3:
annotated_bgr = cv2.cvtColor(annotated, cv2.COLOR_RGB2BGR)
else: annotated_bgr = annotated
save_path = "static/output/predicted.jpg"
os.makedirs(os.path.dirname(save_path), exist_ok=True)
cv2.imwrite(save_path, annotated_bgr)
return {"result_image": save_path,
"counts": dict(class_counts),
"area": int(total_area_pixels)}
```

На початковому етапі роботи функції завантажуються ваги моделі у вигляді словника, де кожен ключ відповідає певній моделі з декількох доступних до використання. Далі за допомогою бібліотеки OpenCV зчитується оригінальне зображення у BGR-форматі. Якщо потрібний файл не знайдено, то породжується виняток з повідомленням про помилку. Якщо все добре, то до зображення використовується обрана модель із заданим порогом довіри 0.5, що є розповсюдженою практикою, але може бути змінено за потреби, та на CPU-пристрої. Результатом роботи моделі стає об’єкт, що містить інформацію про виявлені рамки, маски й класи пошкоджень.

На наступному кроці відбувається підраховування кількості об’єктів кожного класу та загальної площі сегментованих областей. Для цього використовується лічильник counter, який інкрементується для кожної

знайденої рамки, а піксельні площі масок акумулюються у змінній total_area_pixels.

Після отримання всієї необхідної статистики, викликається метод plot для генерації анотованого зображення з накладеними масками, яке за замовчуванням повертається в RGB-форматі. Оскільки подальше збереження виконується з використанням OpenCV у BGR-форматі, використовується функція, яка виконує зміну кольорів у зворотному напрямку (RGB→BGR) у випадку коли вхідний масив має три канали.

Після завершення всіх попередніх кроків відбувається збереження отриманого зображення з передбаченнями. В результаті роботи функція формує та повертає словник із трьома ключами: шлях до збереженого зображення, статистика порахованих об’єктів за класами та загальна площа сегментів у пікселях. Таким чином, модуль дає зручний спосіб отримати зображення з результатами аналізу та детальну інформацію про кількість і розмір знайдених пошкоджень.

3.5 Модуль розрахунку оцінок для будівлі

Модуль розрахунку оцінок складається з трьох взаємопов’язаних підмодулів, які послідовно визначають культурну, функціональну цінності та ступінь пошкодження будівлі. Кожний підмодуль реалізовано у вигляді окремої функції, що приймає на вхід метадані об’єкта і повертає числову оцінку в діапазоні від 1 до 10. В результаті ці оцінки потрапляють у єдиний словник scores, що надалі використовується модулем ранжування. Код функцій розрахунку наведено в додатку А.

Функція оцінки культурної цінності аналізує інформацію про статус охоронного об’єкта, дату будівництва, архітектурний стиль і архітектора будівлі. Спочатку перевіряється належність будівлі до списку ЮНЕСКО, національних або місцевих пам’яток, причому найвищий бал надається об’єктам світової спадщини, середній – пам’яткам національного значення,

нижчий – місцевого або нерозділеного статусу. Далі відбувається аналіз року будівництва, що дозволяє додати додаткові бали залежно від віку споруди: найстаріші об’єкти мають вищий пріоритет за культурною важливістю. Архітектурний стиль перевіряється на належність до переліку рідкісних і поширених шкіл, що дає змогу виділити споруди з унікальними, важливими для збереження архітектурними напрямами. Після цього відбувається обробка інформації про архітектора: згадка участі відомих архітекторів при будівництві будівлі надає додаткові бали у оцінку. Після аналізу всієї доступної інформації відбувається формування остаточного результату, який коригується в залежності від типу будівлі – культурні об’єкти отримують додаткові бали, а звичайні житлові будинки, побудовані за типовими проєктами, навпаки, знижують підсумок. Отримане значення нормалізується відносно максимального можливого та округлюється до цілого, з обмеженням у межах від одиниці до десяти.

Функціональна оцінка базується на аналізі призначення споруди та її поточного стану. Спочатку аналізується ключове інформаційне поле «мета/назва» будівлі, щоб визначити її тип: житлова, госпітальна, навчальна, комерційна чи індустріальна. Кожному типу відповідає базова кількість балів, а наявність супутньої комерційної функції (наприклад, житло з магазином) додає незначну перевагу звичайному житловому будинку. Далі додатково враховується поточний технічний стан споруди – будівля, що експлуатується, отримує додаткові бали, тоді як занедбані або покинуті споруди навпаки знижують підсумок, бо не є пріоритетними у відновленні. У випадку житлових комплексів та приватних будинків за наявності даних про кількість квартир враховується масштаб об’єкта: великі комплекси отримують додаткові бали, враховуючи те, що їх використовує значна кількість людей. Наприкінці, до отриманої суми балів додається остаточна корекція за типом споруди, після чого результат нормалізується до шкали від 1 до 10.

Функція оцінки пошкоджень спирається на зведення результатів роботи модуля комп’ютерного зору, який повертає кількість об’єктів кожного класу пошкоджень, та будівельну інформацію, отриману з відкритих джерел. При розрахунку оцінки для кожного типу руйнування передбачено власну вагу, що відображає небезпечність такого пошкодження для стійкості будівлі. Далі до базового внеску кожного типу пошкодження додається корекція з урахуванням кількості знайдених об’єктів такого класу. Для великих чисел застосовуються підвищувальні коефіцієнти, навіть, якщо пошкодження не є серйозним. Якщо доступні додаткові дані про конструктивні особливості будівлі (наприклад, матеріал зовнішніх або несучих стін, наявність технічного поверху тощо), то ці параметри корегують вагу кожного виду ушкоджень відповідно до їхнього впливу на міцність конструкції. Після накопичення сукупного балу пошкодження весь показник нормалізується та масштабуються до загальної шкали, а отриманий результат округлюється і обмежується в межах від 1 до 10.

Таким чином, модуль розрахунку оцінок забезпечує багатовимірний якісний аналіз кожної будівлі з урахуванням історико-культурних, соціально-функціональних і технічних характеристик, формуючи вхідні параметри для моделі багатокритеріального ранжування.

3.6 Модуль ранжування об'єктів

Модуль ранжування об'єктів є критично важливим для визначення пріоритетності відновлення будівель. Як вже зазначалось раніше, в якості алгоритму ранжування було використано алгоритм TOPSIS. В рамках даного модулю було реалізовано основну функцію topsis_rank та допоміжну функцію _to_number. Допоміжна функція _to_number потрібна лише для коректного перетворення вхідного значення оцінки на ціле число, якщо воно таким не є, уникаючи аварійного завершення програми за рахунок некоректних або відсутніх даних.

Головною функцією, яка й виконує ранжування є функція topsis_rank, яка приймає на вхід список будівель, кожна з яких представлена словником з вкладеним підсловником scores, що містить значення трьох критеріїв: ступені пошкодженості, соціального значення та культурно-історичної цінності. Додатково передаються два параметри – вектори критеріїв ідеального та антиідеального прикладу. Вектор ідеального прикладу містить ті значення критеріїв, які вважаються найкращими з погляду пріоритетності відновлення об’єкта, а антиідеальний – відповідно, найгірші. Програмний код такої функції наведено в лістингу 3.2.

Лістинг 3.2 – Програмний код функції функції ранжування

```
def topsis_rank(buildings, ideal=(4, 10, 5), anti_ideal=(8,
1, 1)):
for b in buildings:
sc = b.get("scores", {})
d_val = _to_number(sc.get("damage", 0))
f_val = _to_number(sc.get("functional", 0))
c_val = _to_number(sc.get("cultural", 0))
vec = (d_val, f_val, c_val)
d_pos = math.sqrt(sum((vec[i] - ideal[i])**2 for i in
range(3)))
d_neg = math.sqrt(sum((vec[i] - anti_ideal[i])**2 for i in
range(3)))
score = d_neg / (d_pos + d_neg) if (d_pos + d_neg) else 0.0
b["topsis_score"] = round(score, 4)
buildings.sort(key=lambda x: x["topsis_score"],
reverse=True)
for idx, b in enumerate(buildings, start=1):
b["priority_rank"] = idx
return buildings
```

Перед початком використання системи було вирішено визначити значення критеріїв для найкращого та найгіршого випадків наступним

чином: будівля, яку слід відновити найближчим часом повинна мати найбільше соціальне значення (10 з 10 можливих), при цьому її культурне значення повинно бути середнім (5 з 10), тому що маленьке значення буде не відображати культурної цінності, а велике історичне значення скоріш за все буде означати наявність об’єкту в списках культурного значення, що накладає обмеження на реставрацію та потребує особливих технік проведення відновлення будівлі, що є коштовним і не завжди можливим на практиці. Найоптимальний ступень пошкодження будівлі повинен дорівнювати 3, тому що надто мале значення може позначати лише пошкодження вікон та незначні пошкодження фасаду, що не несе загрози для стійкості будівлі і отже будівля може ще деякій час використовуватись без обмежень; велике значення пошкодження сигналізує про можливу потребу в зносі будівлі, а отже не потребує проведення відновлювальних робіт. Найгіршим випадком, будівлею, яку можна відновлювати в останню чергу є соціально незначуща споруда, що не має культурного значення. Ступінь пошкодження такої будівлі становить 9, що каже про значні пошкодження будівлі і скоріш за все не підлягає відновленню. Так як така будівля не є пам’ятником культури, то її можна легко знести, а отже не потрібно відновлювати.

У процесі роботи функції topsis_rank для кожного об'єкта окремо виконуються наступні кроки: отримання значення трьох оцінок з підсловника scores; обробка кожного значення критеріїв допоміжною функцією _to_number для забезпечення стабільності алгоритму і виключення можливості виникнення помилки через відсутність або некоректний формат даних; формування вектору оцінок. Далі відбувається обчислення відстані між вектором поточної будівлі та ідеальним вектором, а також між таким вектором і антиідеальним вектором. Таким чином визначається наскільки близько/далеко об'єкт, що розглядається, знаходиться до ідеального або, навпаки, до найгіршого можливого варіанту. На основі цих двох відстаней розраховується агрегований індекс близькості

до ідеального рішення. Цей індекс набуває значення в межах від 0 до 1, де значення, близьке до 1, означає, що будівля ближча до ідеального варіанту і, відповідно, має вищий пріоритет у ранжуванні.

Після обчислення індексу для кожної будівлі, список зі всіма записами будівель сортується у порядку спадання цього індексу, що означає, що найпріоритетніші будівлі опиняються на початку списку. Додатково кожному об’єкту присвоюється поле priority_rank, яке визначає його порядковий номер у відсортованому списку, починаючи з 1 для об'єкта з найвищим індексом.

Отже, у результаті виконання зазначеної функції, для кожної будівлі з вхідного списку додаються два додаткові значення: topsis_score, що містить числовий показник близькості до ідеального рішення, та priority_rank, який вказує на позицію будівлі у загальному рейтингу пріоритетності. Функція повертає оновлений список будівель, впорядкований за значущістю, що далі виводиться для ознайомлення користувачеві. Такий підхід, що реалізований функцією topsis_rank, дозволяє приймати більш обґрунтовані рішення щодо черговості відновлення будівель, враховуючи не лише їх технічний стан, а й їхню соціальну і культурну значущість.

3.8 Серверна частина та API

Серверна компонента застосунку виконана як Flask-додаток і виступає єдиною точкою входу всіх клієнтських запитів. Вона організовує маршрутизацію HTTP-запитів, координує виклики всіх потрібних функцій, отримання і підготовки даних з зовнішніх сервісів, розрахунку оцінок і їхнього збереження. Далі наведено огляд основних маршрутів і їх логіки, код, яких представлено в додатку Б.

Головний маршрут за шляхом «/» потрібен для обробки GET-запиту та читання файлу saved_buildings.json, у якому знаходятся збережені записи про будівлі з попередніми результатами аналізу. Якщо такий файл існує, то

його вміст десеріалізується у список об’єктів, інакше відбувається формування порожнього масиву. На наступному кроці такий список передається у шаблон list.html, що відповідає за відображення табличного переліку будівель з їх адресами та раніше обчисленими оцінками.

GET-запит за шляхом /add просто повертає порожню форму, реалізовану в шаблоні index.html, де користувач може завантажити фото й вводити адресу будівлі.

Після надсилання зазначеної форми формується POST-запит на маршрут /upload. На сервер передається файл-зображення та рядок із повною адресою будівлі, після чого в функції для обробки такого маршруту відбувається перевірка на наявність та зберігання зображення у папці static/uploads. Далі викликається функція predict_and_visualize_return_data, яка відповідає за роботу попередньо натренованої моделі виявлення пошкоджень на завантаженому користувачем зображенні, і повертає шлях до збереженого анотованого зображення, лічильник виявлених класів пошкоджень та загальну площу масок. Для зручності зазначені дані тимчасово зберігаються в сесії Flask.

На наступному кроці з отриманої стрічки з адресою будівлі виділяються номер будинку та назва вулиці. У разі невдачі парсингу користувачам надсилається повідомлення про помилку виявлення адреси. У разі вдалого виконання такої операції відбувається запуск звернення до зовнішнього джерела (web scraping) через функцію get_building_info, яка повертає словник ключів з основною та додатковою інформацією про будівлю, отриманої з відкритих джерел інформації. Отримані поля обох видів інформації фільтруються за заздалегідь визначеним переліком, залишаючи лише релевантні атрибути. Якщо в результаті роботи зовнішнього пошуку з’являється інформація про серію і проект будівлі, то викликається функція get_project_info, яка виконує пошук та отримання необхідної інформації про типові серії та проєкти будинків, і в якості результату повертає словник з отриманими даними, що також зберігаються

в сесії. Через функцію find_monument_info здійснюється пошук даних щодо статусу історично-культурної пам’ятки.

Після підготовки всіх необхідних даних користувача перенаправляється на маршрут /info, де у шаблоні info.html відображаються: анотоване фото, лічильник пошкоджень, основна інформація про будівлю та додаткова, що показує функціональну роль споруди, інформація про серію та проєкт.

Маршрут /scores реалізовано як API-метод, що повертає результати трьох функцій оцінки – estimate_damage_severity, estimate_functional_value та estimate_cultural_value. Він дістає з сесії раніше збережені необхідні дані та передає їх відповідним підмодулям, повертаючи JSON-об’єкт з трьома числовими полями: damage, functional і cultural. В рамках даного маршруту реалізовано асинхронне оновлення значень оцінок будівлі на клієнтській стороні без перезавантаження веб-сторінки.

Для збереження отриманої інформації про будівлю та обчислених оцінок користувач робить POST-запит на маршрут /save. В такому разі сервер читає з сесії адресу і статистику пошкоджень, а з тіла запиту остаточні значення трьох оцінок, а далі формує новий запис останнім в файл saved_buildings.json. Після цього відповідає об’єктом із статусом «ok», підтверджуючи успішне збереження.

GET-маршрут /list дублює функціонал головної сторінки, але призначений для відображення лише вже збережених записів. Маршрут /topsis запускає повторне ранжування всіх записів у разі надходження від користувача нового запису про будівлю. Після виконання ранжування сесія виконує перезапис файлу saved_buildings.json з урахуванням нового порядку відновлення і перенаправляє користувача назад на /list, де відображається оновлений рейтинг.

Підбиваючи підсумки, можна побачити, що серверна частина розробленого застосунку на Flask забезпечує повний життєвий цикл роботи з даними: від завантаження та аналіз зображень, через обробку зовнішніх

даних й обчислення оцінок, до їх збереження та багатокритеріального ранжування. Така архітектура дозволяє легко масштабувати функціонал та інтегрувати нові сервіси чи модулі без значної перебудови коду.

3.9 Інтерфейс користувача

При завантажені застосунку, користувач відразу має можливість побачити список будівель, які були пошкоджені в ході воєнних дій та потребують відновлення, у вигляді вже відсортованого списку в порядку від будівель, що потребують найшвидшого втручання до тих, які можуть зачекати надходження нових інвестицій. На рисунку 3.7 представлено вигляд головної сторінки застосунку.

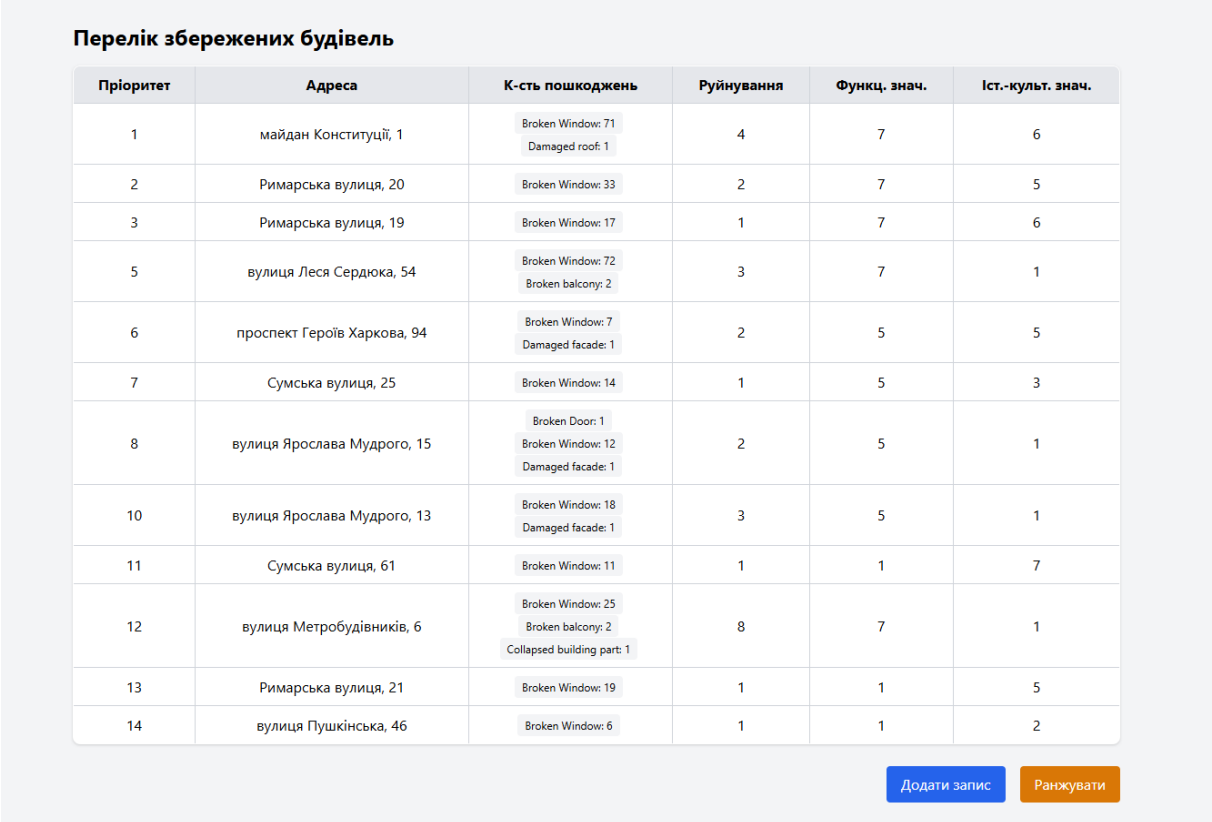


Рисунок 3.7 – Головна сторінка застосунку

Після ознайомлення зі списком пошкоджених будівель, користувач має можливість додати новий запис – зареєструвати нову зруйновану будівлю, натиснувши на кнопку «Додати запис». Після натискання на зазначену кнопку користувач перенаправляється на сторінку для додавання запису, що має вигляд представлений на рисунку 3.8.

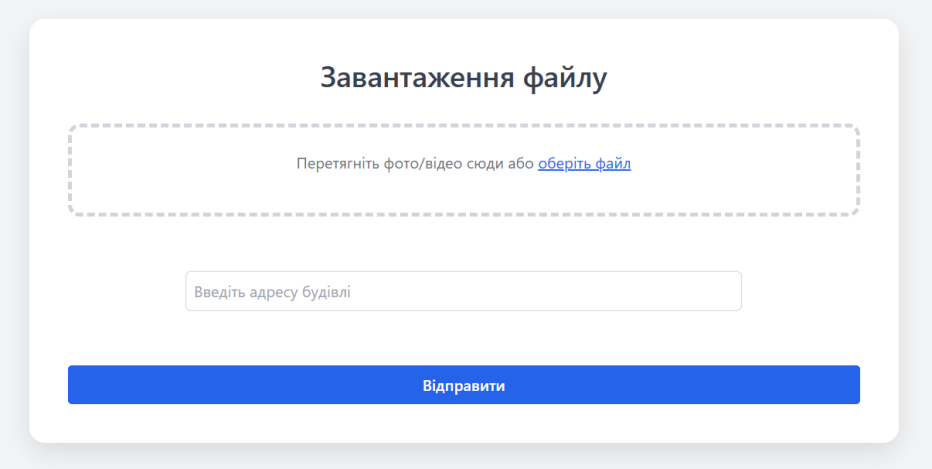


Рисунок 3.8 – Сторінка з формою для додання нового запису

Як можна побачити на рисунку, на даній сторінці представлено форму для внесення необхідних даних: адреси будівлі та завантаження одного або декількох фото будівлі з різних ракурсів. Після натискання користувачем кнопки «Відправити» відбувається обробка завантаженої інформації та надсилання її на сервер. За вказаною адресою відбувається пошук загальної інформації про будівлю, інформації про типовий проєкт, за яким її було побудовано, перевірка наявності адреси в реєстрах пам’яток місцевого та національного значення. Паралельно з цим відбувається обробка завантажених зображень для автоматичного виявлення пошкоджень. Статус виконання таких процесів відображається на сторінці з формою для зручності користувача та можливості відстеження процесу. На рисунку 3.9 представлено вигляд такої сторінки після завантаження користувачем всіх необхідних даних та відправки їх на сервер.

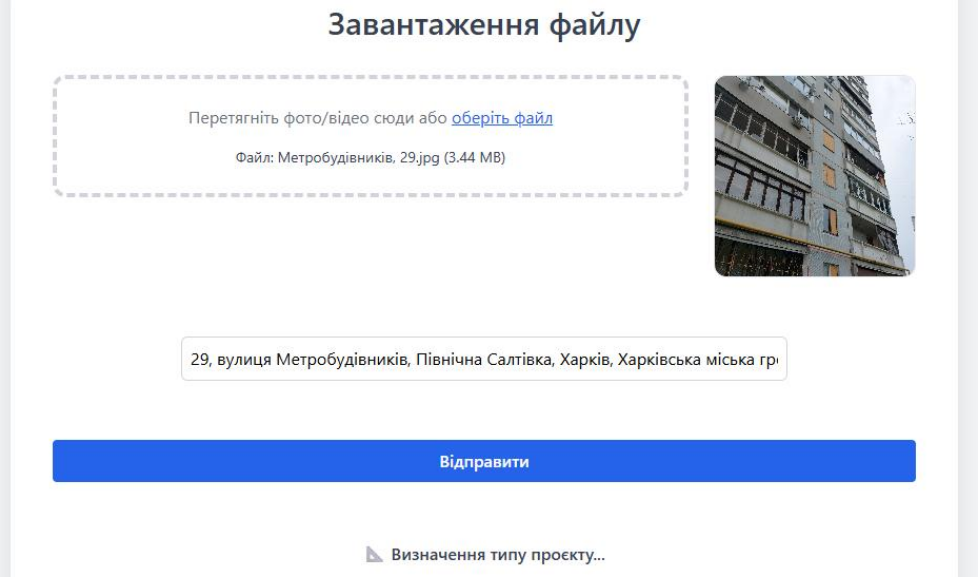


Рисунок 3.9 – Вигляд сторінки з формою для додання нового запису після її відправки

Після отримання всієї необхідної інформації користувача перенаправляється на сторінку, вигляд якої представлено на рисунку 3.10 – 3.11, зі всією зібраною інформацією про будівлю, з якою користувач може ознайомитись.

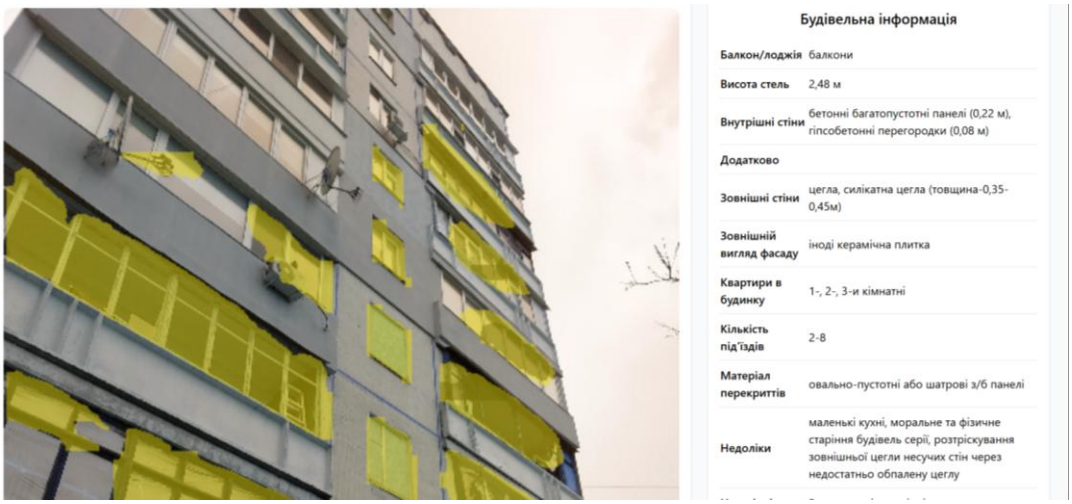


Рисунок 3.10 – Перша частина сторінки зі зібраною інформацією про будівлю

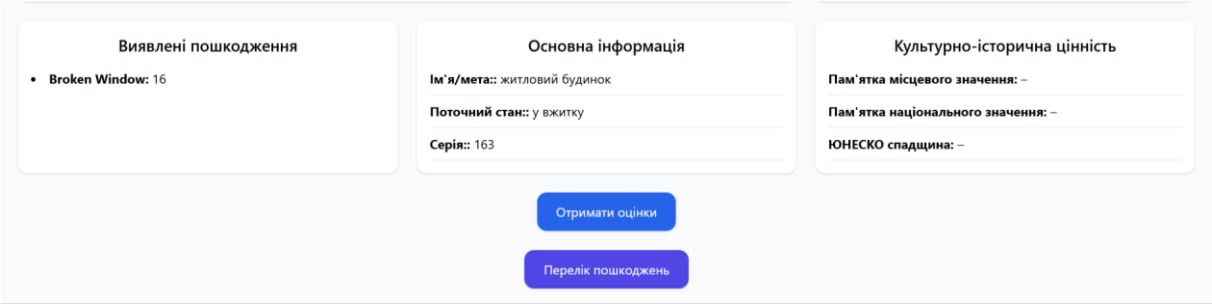


Рисунок 3.11 – Друга частина сторінки зі зібраною інформацією про будівлю

Після перегляду інформації про будівлю користувач має можливість розрахувати необхідні для проведення ранжування оцінки натисканням на кнопку «Отримати оцінки».

У спливаючому вікну, представленого на рисунку 3.12 відображаються автоматично обчислені оцінки та коментарі для їхнього розуміння користувачем. Користувач має можливість зберегти такі значенням натисканням на відповідну кнопку у вікні.

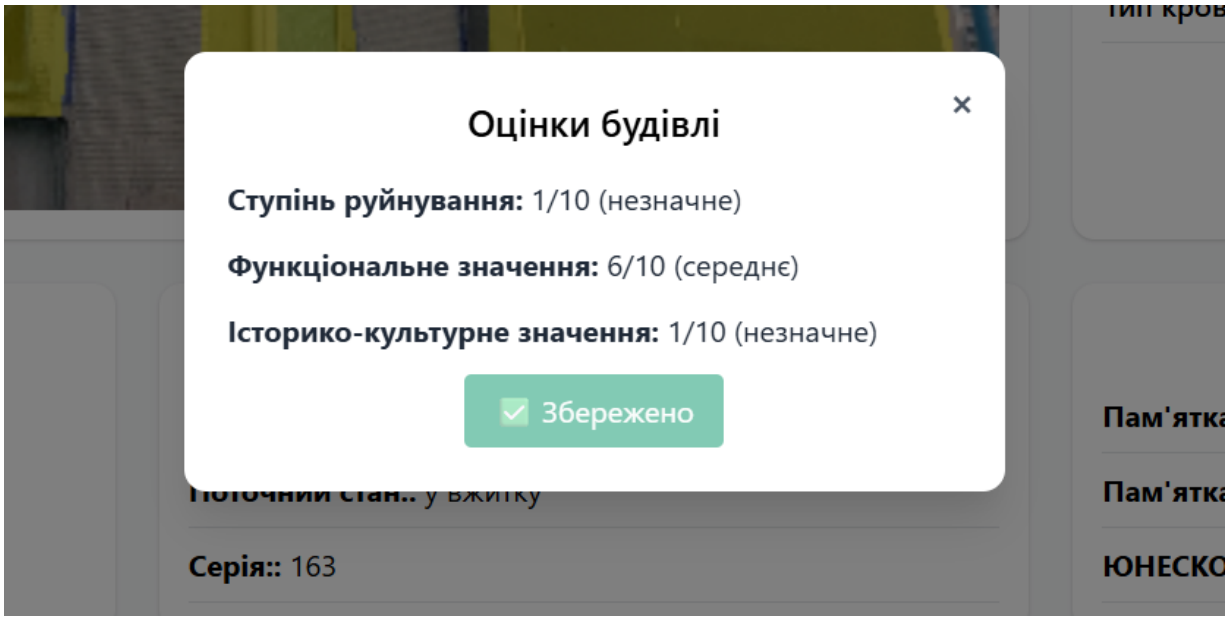
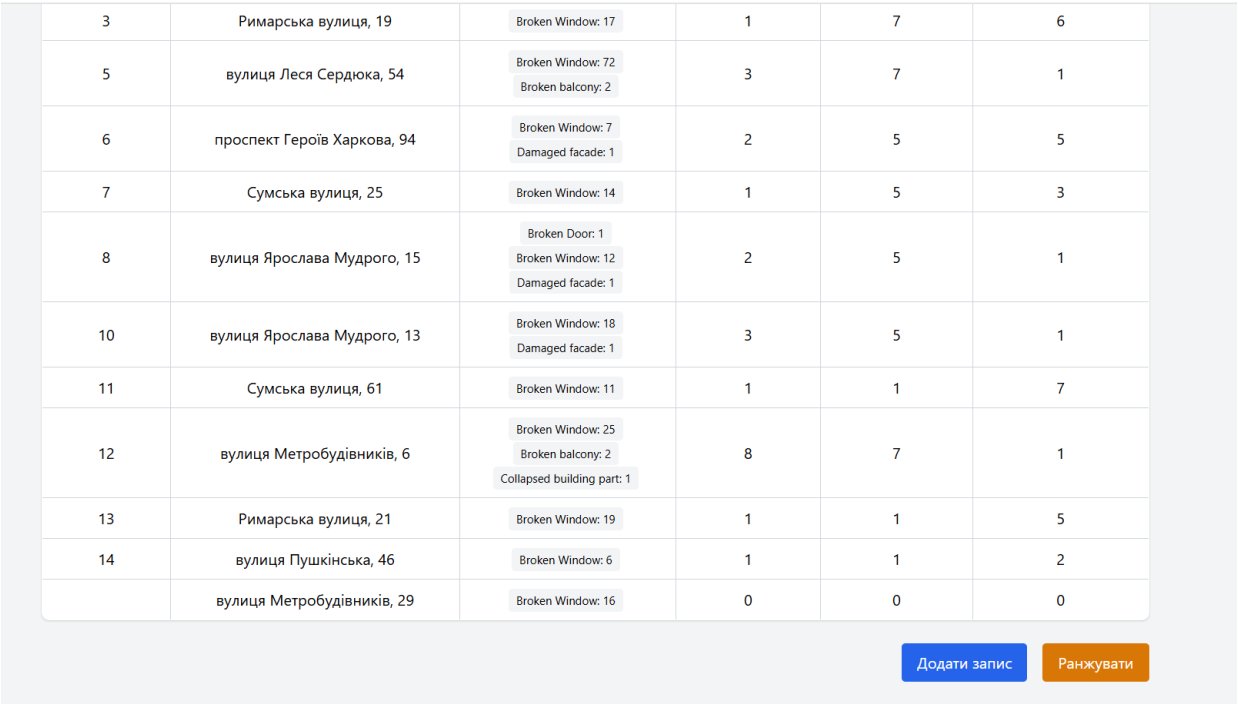


Рисунок 3.12 – Спливаюче вікно з обчисленими для будівлі оцінками після збереження інформації

Після того, як запис про будівлю з її адресою та оцінками було збережено, користувач має можливість перейти на сторінку зі списком будівель і переконатись, що будівлю було додано до переліку зареєстрованих відновлених будівель, але для неї ще не було обчислено пріоритету відновлення. Вигляд сторінки зі списком будівель у описаному стані відображено на рисунку 3.13.



3	Римарська вулиця, 19	Broken Window: 17	1	7	6
5	вулиця Леся Сердюка, 54	Broken Window: 72 Broken balcony: 2	3	7	1
6	проспект Героїв Харкова, 94	Broken Window: 7 Damaged facade: 1	2	5	5
7	Сумська вулиця, 25	Broken Window: 14	1	5	3
8	вулиця Ярослава Мудрого, 15	Broken Door: 1 Broken Window: 12 Damaged facade: 1	2	5	1
10	вулиця Ярослава Мудрого, 13	Broken Window: 18 Damaged facade: 1	3	5	1
11	Сумська вулиця, 61	Broken Window: 11	1	1	7
12	вулиця Метробудівників, 6	Broken Window: 25 Broken balcony: 2 Collapsed building part: 1	8	7	1
13	Римарська вулиця, 21	Broken Window: 19	1	1	5
14	вулиця Пушкінська, 46	Broken Window: 6	1	1	2
	вулиця Метробудівників, 29	Broken Window: 16	0	0	0

Додати запис Ранжувати

Рисунок 3.13 – Вигляд сторінки зі списком будівель після створення запису для нової будівлі

Для врахування доданої будівлі в списку з пріоритетами відновлення користувачу необхідно натиснути на кнопку «Ранжувати». Після натискання такої кнопки список будівель буде оновлено з урахуванням щойно доданого запису про нову будівлю.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Аналіз моделей і методів боротьби з дисбалансом класів

Для оцінювання ефективності моделей детекції та сегментації об'єктів в умовах сильного дисбалансу класів у наборі даних було навчено дві архітектури – YOLOv11 і Mask R-CNN (реалізація через фреймворк Detectron2). Навчання моделей проводили із застосуванням кількох стратегій, спрямованих на поліпшення якості розпізнавання об'єктів із рідкісних класів, а саме: базовий підхід без змін, аугментація даних, використання зваженого завантажувача даних (weighted dataloader) і комбінація аугментації з weighted dataloader. В даному розділі спочатку наводяться особливості тренування та налаштування обраних моделей сегментації екземплярів, далі проводиться порівняння якості виявлення пошкоджень різних типів та розмірів зазначеними моделями, враховуючи різні підходи боротьби з дисбалансом класів.

Відразу варто зазначити, що для роботи з обома згаданими архітектурами завантаження зображень відбувалося через Roboflow API у потрібному для моделей форматі. Навчання моделей проводилося в хмарному середовищі Colab Notebook. На лістингу 4.1 представлено код для завантаження потрібних версій набору даних та формування системного файлу з інформацією про завантажений набір даних.

Лістинг 4.1 – Програмний код для завантаження набору даних

```
project = rf.workspace("nataliia-zh9ay").project("war-
damage-segmentation")
version = project.version(10)
dataset = version.download("yolov11")
data_yaml_path = "war-damage-segmentation-10/data.yaml"
with open(data_yaml_path, 'r') as f:
data_yaml = yaml.safe_load(f)
```

Продовження лістингу 4.1

```
data_yaml['train'] = f"{dataset.location}/train/images"
data_yaml['val'] = f"{dataset.location}/valid/images"
data_yaml['test'] = f"{dataset.location}/test/images"
with open(data_yaml_path, 'w') as f:
yaml.dump(data_yaml, f)
```

Першою моделлю, яка розглядається в рамках цього розділу, є одна з останніх моделей сімейства YOLO – YOLOv11 з вагами для сегментації. Навчання цієї моделі проводилося з використанням стандартної конфігурації, адаптованої під нашу задачу, із застосуванням функцій втрат, характерних для цієї архітектури (включно з box loss, classification loss, objectness loss і dfl loss), які вже вбудовані в модель для швидкого й точнішого навчання на наборах зображень зі складними сценами. На рисунку 4.1 наведено код для завантаження ваг та тренування моделі без попереднього налаштування.

```
model_path = "runs/segment/yolov11_seg_war_damage/weights/best.pt"

from google.colab import files
files.download(model_path)
```

```
model = YOLO("yolo11n-seg.pt")
```

```
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolo11n-seg.pt to 'yolo11n-seg.pt'...
100%|██████████| 5.90M/5.90M [00:00<00:00, 407MB/s]
```

```
model.train(
    data="war-damage-segmentation-10/data.yaml",
    epochs=50,
    imgsz=1024,
    batch=16,
    device=0,
    patience=20,
    name="yolov11_seg_war_damage"
)
```

```
Ultralytics 8.3.128 🚀 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
engine/trainer: agnostic_nms=False, amp=True, augment=False, auto_augment=randaugment, batch=16, bgr=0.0, box=7.5, cache=False, cfg=None, classes=None,
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 83.0MB/s]Overriding model.yaml nc=80 with nc=10

                   from  n    params  module                                       arguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv             [3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv             [16, 32, 3, 2]
  2                  -1  1      6640  ultralytics.nn.modules.block.C3k2            [32, 64, 1, False, 0.25]
  3                  -1  1     36992  ultralytics.nn.modules.conv.Conv             [64, 64, 3, 2]
```

Рисунок 4.1 – Код завантаження ваг та тренування YOLOv11

Для реалізації другої обраної для дослідження моделі: Mask R-CNN була використана бібліотека Detectron2, що надає зручний інтерфейс для навчання та оцінювання різних моделей комп'ютерного зору. Цю модель також було завантажено та налаштовано з базовими параметрами. Базове налаштування моделі та її навчання представлено на лістингу 4.2.

Лістинг 4.2 – Код для налаштування та тренування Mask R-CNN

```
ARCHITECTURE = "mask_rcnn_R_101_FPN_3x"
CONFIG_FILE_PATH = f"COCO-
InstanceSegmentation/{ARCHITECTURE}.yaml"
MAX_ITER = 2000
EVAL_PERIOD = 200
BASE_LR = 0.001
NUM_CLASSES = 11
OUTPUT_DIR_PATH = os.path.join(
DATA_SET_NAME,
ARCHITECTURE,
datetime.now().strftime('%Y-%m-%d-%H-%M-%S'))
os.makedirs(OUTPUT_DIR_PATH, exist_ok=True)
cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file(CONFIG_FILE_
PATH))
cfg.MODEL.WEIGHTS =
model_zoo.get_checkpoint_url(CONFIG_FILE_PATH)
cfg.DATASETS.TRAIN = (TRAIN_DATA_SET_NAME,)
cfg.DATASETS.TEST = (TEST_DATA_SET_NAME,)
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 64
cfg.TEST.EVAL_PERIOD = EVAL_PERIOD
cfg.DATALOADER.NUM_WORKERS = 2
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.INPUT.MASK_FORMAT='bitmask'
cfg.SOLVER.BASE_LR = BASE_LR
cfg.SOLVER.MAX_ITER = MAX_ITER
cfg.MODEL.ROI_HEADS.NUM_CLASSES = NUM_CLASSES
```

Продовження лістингу 4.2

```
cfg.OUTPUT_DIR = OUTPUT_DIR_PATH
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

Для боротьби з дисбалансом класів було розглянуто різні підходи до вирішення такої проблеми. Першим підходом стала аугментація даних, яка містила як перетворення, спрямовані на збільшення різноманітності даних, як-от додавання шуму, зміна яскравості та контрасту, випадкове масштабування, так і випадкове нарізання зображень. перший різновид аугментації дав змогу урізноманітнити вхідні зображення, збільшивши їхню кількість у наборі даних і змушуючи моделі ставати стійкішими до різної якості зображень і розмаїття форматів зйомки. Другий підхід дав змогу моделі звертати увагу на дрібні об'єкти та об'єкти, що перебувають у меншості, які на вихідних великих зображеннях займали мало площі для ефективного розпізнавання. Цей підхід не вимагав додаткового налаштування процесів навчання моделей, а був реалізований як завантаження іншої версії набору даних із внесеними в нього змінами.

Як другий підхід до вирішення цієї проблеми було вирішено використовувати weighted dataloader - завантажувач даних, що реалізує вибірку навчальних прикладів з урахуванням ваги класів. Такий підхід дав змогу збільшити ймовірність потрапляння в батч зображень із рідкісними об'єктами. Цей підхід було реалізовано на першій версії набору даних, яку не було аугментовано, і він зажадав додавання в основний процес навчання моделей додавання класу YOLOWeightedDataset [56] для моделі YOLOv11 і класу CustomTrainer для Mask R-CNN, код якого наведено в додатку В.

Останнім способом стала комбінація аугментації даних і weighted dataloader. Для реалізації цього підходу моделі було натреновано на аугментованому наборі даних та з використанням щойно зазначених класів для завантаження зображень с урахуванням ваг. Отримані оцінки mAP@0.5

та mAP@0.5:0.95 для кожної комбінації моделі зі способом боротьби з дисбалансом класів наведено у таблиці 4.1.

Таблиця 4.1 – Оцінки ефективності виявлення пошкоджень моделей з різними підходами до боротьби з дисбалансом класів

	Модель	Підхід для вирішення проблеми дисбалансу класів			
		Без налаштувань	Аугментація даних	Weighted dataloader	Комбінація підходів
mAP@0.5	YOLOv11	0.65	0.74	0.89	0.89
mAP@0.5:0.95		0.31	0.37	0.43	0.43
mAP@0.5	Mask-R-CNN	0.61	0.73	0.90	0.91
mAP@0.5:0.95		0.29	0.35	0.44	0.45

Аналіз наведених результатів обчислення метрик mAP@0.5 та mAP@0.5:0.95 показує, що застосування методів боротьби з дисбалансом класів значно підвищує якість передбачень обох моделей. Аугментація даних забезпечує помітний приріст порівняно з базовим навчанням, що пояснюється поліпшенням узагальнювальної здатності моделей і збільшенням різноманітності навчальних прикладів. Використання weighted dataloader показує подальше поліпшення метрик, зумовлене більш збалансованим навчанням, за якого моделі приділяють більше уваги рідкісним класам.

Для Mask R-CNN використання комбінації аугментації та weighted dataloader дозволило отримати найкращі результати, як для цієї моделі, так і серед всіх проведених експериментів. У випадку YOLOv11 комбінування цих методів не призвело до значного поліпшення метрики mAP порівняно з використанням тільки weighted dataloader, можливо, через уже високу якість балансування при застосуванні останнього.

Загалом, представлені в таблиці результати підтверджують, що запропоновані шляхи вирішення проблеми дисбалансу класів є ефективними інструментом та допомагають значно покращити точність виявлення пошкоджень обох моделей.

Для наочності результатів виявлення пошкоджень моделями наводяться колажі з 4 зображеннями. Верхній ряд – результати роботи моделі YOLOv11, нижній ряд – робота моделі Mask-R-CNN. Зображення, що представлені зліва, є найгіршими результатами, що надали моделі (звичайний підхід без використання методів боротьби з дисбалансом класів), справа представлені найкращі результати роботи моделей (комбінація підходів аугментації даних та використання weighted dataloader). Для відображення найбільшого покращення результатів моделей було обрані зображення, що спочатку були проблемними для опрацювання. На рисунку 4.2 показано вплив рішення проблеми дисбалансу класів на можливість моделей виявлення пошкоджень.

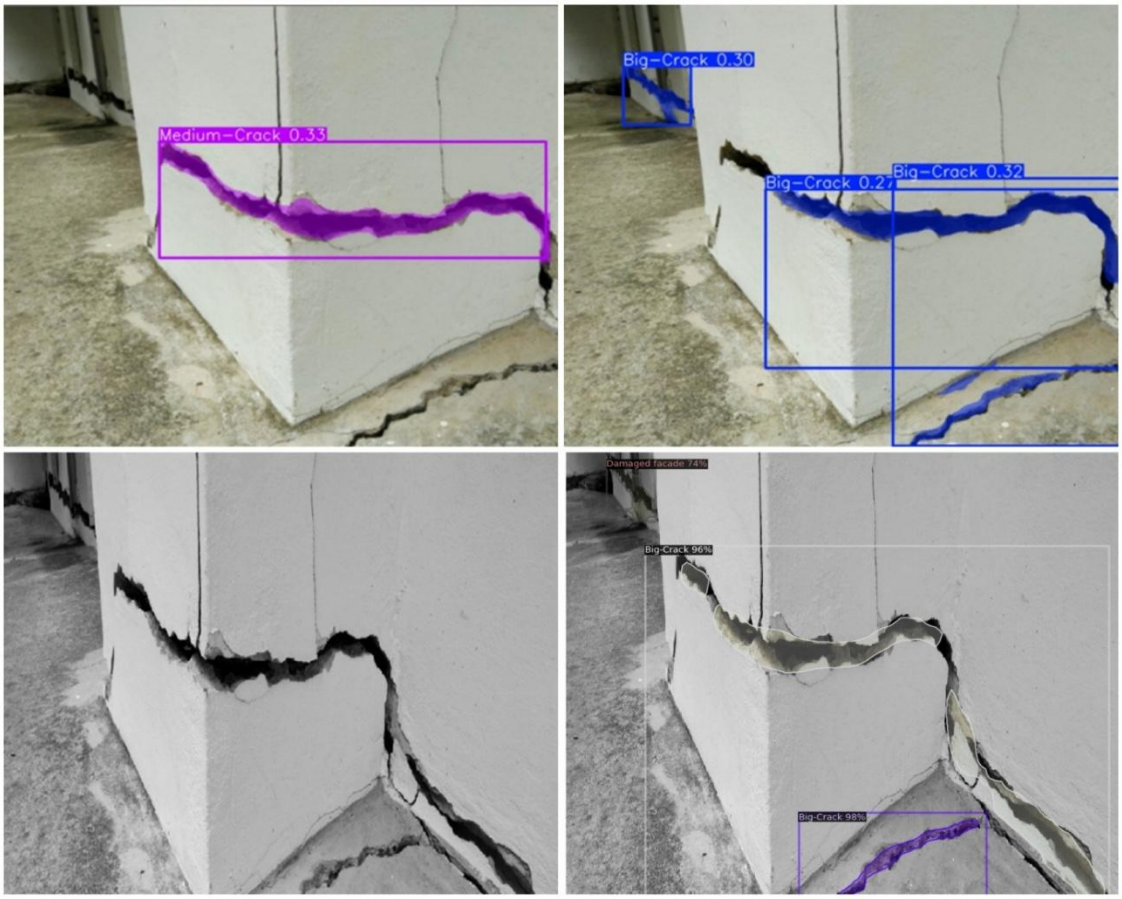


Рисунок 4.2 – Результати виявлення моделями YOLOv11 та Mask- R-CNN трещин до та після використання комбінованого підходу борьби з дисбалансом класів

Як можна побачити на наведеному колажі, YOLO спочатку без необхідного налаштування змогла виявити тріщину, в той час, коли Mask- R-CNN не виявила нічого. Після завершення тренувань з використанням комбінації підходів для вирішення проблеми дисбалансу класів, точність виявлення пошкоджень моделями змінилась. Обидві моделі змогли виявити всі пошкодження, однак Mask- R-CNN показала набагато кращі результати: чіткіше виявивши межі пошкоджень та зробивши це з більшою ймовірністю.

На рисунку 4.3 представлені для порівняння результати роботи моделей на зображені віддаленою будівлі з великою кількістю екземплярів мажорного класу, та присутністю екземпляру мінорного класу.

Як можна побачити моделями було надано зовсім різні результати. Після тренування моделей з базовими налаштуваннями YOLOv11 виявила лише пошкоджені вікна у будинку, тобто виявила екземпляри мажорного класу. Mask- R-CNN правильно виявила межі для мінорного класу, але визначила неправильний клас пошкодження, та не виявила екземпляри мінорного класу, що я занадто маленькими на зображені – мають меншу кількість пікселів. Після навчання моделей з урахуванням дисбалансу класів, обидві моделі значно покращили свої результати. YOLOv11 змогла виявити мінорний клас, та правильно визначити його межі. Mask- R-CNN правильно визначила мінорний клас та почала виявляти найбільш явні пошкодження вікон.

Попри те, що YOLO модель у цьому разі змогла розпізнати дрібніші за розміром ушкодження і виявити сумарно більше екземплярів, явну перемогу їй віддати складно. У даному випадку варто враховувати, що помилка не виявлення пошкодженого вікна є менш серйозною, ніж не виявлення або виявлення з меншою впевненістю більш серйозного класу ушкоджень, які, як зазначалося при описі тренувального набору зображень, є мінорними. Варто зазначити, що Mask- R-CNN в обох випадках виконувала визначення пошкоджень з більшою впевненістю.

Рисунок 4.3 – Результати виявлення пошкоджень моделями YOLOv11 та Mask- R-CNN на складних сценах до та після використання комбінованого підходу борьби з дисбалансом класів

Отже, після проведення візуального порівняння результатів роботи моделей та аналізу отриманих метрик, можна зробити висновок про позитивний вплив використання методів боротьби з дисбалансом класів, особливо на зображеннях будівель із проблемними для виявлення для моделей пошкодженнями. Загалом обидві моделі продемонстрували високі показники якості, що підтверджує їхню придатність для практичного

застосування в задачах виявлення пошкоджень. Отримані результати показують, що як YOLOv11, так і Mask R-CNN ефективно справляються із завданням за умови застосування підходів до боротьби з дисбалансом класів. З огляду на відмінності в архітектурі та специфіці оброблення зображень, доцільно надати користувачеві можливість вибору моделі залежно від типу пошкоджень і умов аналізу. Такий підхід дасть змогу адаптувати систему під конкретні сценарії та підвищити точність результатів у реальних умовах експлуатації.

4.2 Перспективи розвитку

У перспективі стратегічним напрямком розвитку описаної в рамках кваліфікаційної роботи ідеї та розробленого прототипу системи є побудова комплексної автоматизованої системи, що у загальному вигляді представлена на рисунку 3.1. Така система може забезпечити інтеграцію багаторівневих даних для ефективного моніторингу та аналізу пошкоджень будівель, що виникли внаслідок воєнних дій на території України, а також визначення пріоритетів проведення відновлювальних робіт.

Ключовим завданням при розробці зазначеної системи є організація процесу автоматичного виявлення пошкоджених споруд на основі аналізу супутникових і аерофотознімків з використанням сучасних методів комп'ютерного зору. Паралельно з цим планується реалізувати модуль опрацювання та систематизації звітів, що надходять від прокуратури та інших державних органів і обробку звернень волонтерів, що дасть змогу забезпечити комплексний збір інформації, підвищити повноту та достовірність даних про пошкодження будівель. Такий підхід дозволить значно розширити географічне охоплення збору даних і дасть змогу формувати динамічний реєстр пошкоджених споруд, що буде регулярно оновлюватиметься і слугуватиме основою для оперативного реагування і прозорості витрат інвестицій.

Важливим компонентом майбутньої системи також є розробка і впровадження методів побудови тривимірних моделей зруйнованих об'єктів на основі аналізу зображень із дронів та при наземному обстежені. Побудовані 3D-моделі забезпечать високоточне просторове представлення стану будівель, що дозволить істотно спростити експертну оцінку і моніторинг стану будівель без необхідності фізичної присутності фахівців на об'єктах, що в свою чергу знижуючи витрати часу і ресурсів. Крім того, планується активна взаємодія з профільними державними структурами, органами місцевого самоврядування та фахівцями, відповідальними за планування та організацію відновлювальних робіт. У рамках такої взаємодії передбачається отримання доступу до комплексних баз даних з інформацією про архітектурні особливості будівель, їхню історико-культурну цінність, а також технічної документації, що дасть змогу врахувати всі аспекти під час оцінювання серйозності пошкоджень та формування планів відновлення будівель. Аналіз даних, одержуваних під час проведення спеціалізованих інженерних обстежень і досліджень із використанням високоточних інструментальних методів, дасть змогу додатково уточнювати параметри ушкоджень і створювати збагачені датасети для подальшого навчання і валідації моделей.

Останнім, але дуже важливим напрямком майбутньої роботи стане розробка методів автоматизованого формування оптимізованих планів відновлення, заснованих на комплексному аналізі виявлених ушкоджень. Такі плани мають враховувати не тільки технічні параметри руйнувань, а й фінансові, матеріальні та кадрові обмеження. Зокрема, в майбутньому система зможе інтегруватиме дані про доступність і зайнятість ремонтних бригад, особливості технологічного процесу відновлення, наявність і терміни поставок будівельних матеріалів, що дасть змогу формувати пріоритети відновлення будівель з урахуванням багатовимірних чинників, забезпечуючи раціональний розподіл ресурсів і максимальну ефективність проведених робіт.

Зрештою, планується, що реалізована система стане невід'ємним інструментом для підтримки ухвалення рішень на державному та регіональному рівнях про відновлення пошкоджених споруд, сприяючи прискоренню процесів відновлення критично важливої інфраструктури та мінімізації соціально-економічних втрат, пов'язаних із воєнними руйнуваннями.

ВИСНОВКИ

В рамках виконання кваліфікаційної роботі було розроблено концепцію та реалізовано прототип системи з автоматизованим виявленням пошкоджень будівель у міській інфраструктурі на основі методів комп'ютерного зору для визначення пріоритетності відновлення будівель з урахуванням соціального значення, ступеня пошкодження та культурно- історичної цінності. Результат роботи зазначеної системи було протестовано на реальних зображеннях зруйнованих будівель міста Харкова та отримано сформований список пріоритетності відновлення зареєстрованих в системі будівель. Такі результати роботи можуть бути використані в рамках поточних проектів університету, а також у співпраці з державними органами та дослідницькими інституціями, що працюють над проблемами відновлення зруйнованих територій.

Запропонована система є розвитком і адаптацією сучасних міжнародних рішень в галузі автоматичного аналізу пошкоджень інфраструктури, враховуючи специфіку воєнного конфлікту в Україні, та напрямку роботи з алгоритмами багатокритеріального прийняття рішень. Порівняно з існуючими системами система дозволяє поєднати у собі процес збору доступної інформації про будівлю, що цікавить, автоматичне виявлення пошкоджень та визначення пріоритетності її відновлення. Унікальністю даної системи є запропонований підхід визначення першочерговості відновлення будівлі з урахуванням критеріїв її соціального значення, історично-культурної цінності та ступеню пошкодження. Концепція такої системи була описана в тезах доповіді та представлена на мужнародній науково-практичній конференції «Сучасні інформаційні системи та технології в цифровому суспільстві».

В майбутньому планується проводження роботи над висунутою концепцією та перетворення прототипу на повноцінну працюючу систему.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Панченко Л. Зруйнована Україна: наслідки війни для наших міст, природи та сільського господарства. *Хмарочос*. URL: https://hmarochos.kiev.ua/2023/03/27/zrujnovana-ukrayina-naslidky-vijny-dlya-nashyh-mist-pryrody-ta-silskogo-gospodarstva/ (дата звернення: 03.06.2025).

2. Звіт про прямі збитки інфраструктури від руйнувань внаслідок військової агресії Росії проти України станом на початок 2024 року. Київ; 2024. 44 с. URL: https://kse.ua/wp-content/uploads/2024/04/01.01.24_Damages_Report.pdf (дата звернення: 02.04.2025).

3. Damaged cultural sites in Ukraine verified by UNESCO. *Unesco*. URL: https://www.unesco.org/en/articles/damaged-cultural-sites-ukraine-erified-unesco?hub=66116 (date of access: 20.04.2025).

4. Шатрова І.А. Демидова О.О., Мальоркіна І. Оптимізація термінів реконструкції споруд під час війни. *Сучасні досягнення в науці та освіті*: матеріали XIX міжнародної наукової конференції (м. Нетанія, 29 вересня – 6 жовтня 2024 р.). Хмельницький. 2024. С. 107–110.

5. Стало відомо, скільки пошкоджених будинків планують відновити у Харкові. *РЕДПОСТ*. URL: https://redpost.com.ua/news/1280042-stalo-vidomo-skilki-poshkodzhenih-budinkiv-planuyut-vidnoviti-u-harkovi (дата звернення: 14.04.2025).

6. Gupta R. xBD: a dataset for assessing building damage from satellite imagery. 2019. 19 с. DOI: https://doi.org/10.48550/arXiv.1911.09296. (Препринт / arXiv:1911.09296)

7. Building damage assessment in conflict zones: a deep learning approach using geospatial sub-meter resolution data / M. Risso et al. *2025 IEEE 6th international conference on image processing, applications and systems (IPAS)*, Lyon, France, 9–11 January 2025. 2025. P. 1–6. URL: https://doi.org/10.1109/ipas63548.2025.10924534 (date of access:

02.04.2025).

8. Гавриленко К. О., Пилипів В. В. Важливість застосування новітніх технологій та інструментів для відновлення постраждалих міст України. *Програмний комітет*. 2023. С. 255–256.

9. Shevchenko L. Mass housing in ukraine in the second half of the 20th century. *Docomomo journal*. 2022. № 67. URL: https://doi.org/10.52200/docomomo.67.08 (дата звернення: 02.04.2025).

10. Гетун Г.В., Баліна О.І., Безклубенко І.С., Буценко Ю.П., Соломін А.В. Уразливість і ступінь пошкодження будівель та споруд при надзвичайних ситуаціях. *Наука та освіта*: зб. пр. XVII Міжнар. наук. конф., (м. Хайдусобосло, Угорщина, 15–22 січня 2023 р.). 2023. С. 75–80.

11. Ukraine: post-war reconstruction set to cost $524 billion. *UN News*. URL: https://news.un.org/en/story/2025/02/1160466 (date of access: 03.04.2025).

12. Оцінка нерухомих об’єктів культурної спадщини *Оцінка майна*. URL: https://ocinka.ua/oczinka-neruhomyh-obyektiv-kulturnoyi-spadshhyny/#3 (дата звернення: 03.04.2025).

13. Shamoug A. In Ukraine, machine-learning algorithms and big data scans used to identify war-damaged infrastructure. *UNDP*. URL: https://www.undp.org/ukraine/blog/ukraine-machine-learning-algorithms-and-big-data-scans-used-identify-war-damaged-infrastructure (date of access: 11.03.2025).

14. Assessing infra damage in war-torn ukraine. *Geospatial World*. URL: https://geospatialworld.net/prime/technology-and-innovation/assessing-infra-damage-war-torn-ukraine/ (date of access: 11.03.2025).

15. AI – driven building damage assessment: innovations and applications. *Flypix*. URL: https://flypix.ai/blog/building-damage-assessment/ (date of access: 11.04.2025).

16. AI damage detection. URL: https://t2d2.ai/damage-detector (date of access: 13.04.2025).

17. VR museum War up Close. *War up Close.* URL: https://war.city/about-us/ (date of access: 23.04.2025).

18. Ukrainska Pravda. Rebuilding of housing destroyed by Russians can now be seen online in 3D. *Ukrainska Pravda*. URL: https://www.pravda.com.ua/eng/news/2023/10/3/7422472/ (date of access: 23.04.2025).

19. Save ukrainian heritage. *SCAN UA*. URL: https://scanua.com/ (date of access: 24.04.2025).

20. Kashtan V. Y., Hnatushenko V. V. Automated building damage detection on digital imagery using machine learning. *Naukovyi visnyk natsionalnoho hirnychoho universytetu*. 2023. № 6. P. 134–140. URL: https://doi.org/10.33271/nvngu/2023-6/134 (date of access: 15.03.2025).

21. Localizing and quantifying damage in social media images / X. Li et al. *IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*. (Barcelona, 28–31 August 2018). URL: https://doi.org/10.1109/asonam.2018.8508298 (date of access: 18.03.2025).

22. Localizing and quantifying infrastructure damage using class activation mapping approaches / X. Li et al. *Social network analysis and mining*. 2019. Vol. 9, № 1. DOI: https://doi.org/10.1007/s13278-019-0588-4 (date of access: 18.03.2025).

23. Nia K. R., Mori G. Building damage assessment using deep learning and ground-level image data. *2017 14th conference on computer and robot vision (CRV)*. (Edmonton, AB, 16–19 May 2017). 2017. DOI: https://doi.org/10.1109/crv.2017.54 (date of access: 20.04.2025).

24. Klyshnia A. First step towards circular reconstruction. Damage assessment and material mapping of buildings in Ukraine through ground imagery processing. *IAAC BLOG*. URL: https://blog.iaac.net/first-step-towards-circular-reconstruction-damage-assessment-and-material-mapping-of-buildings-in-ukraine-through-ground-imagery-processing/ (date of access: 24.04.2025).

25. Real-Time ground-level building damage detection based on

lightweight and accurate yolov5 using terrestrial images / C. Liu et al. *Remote sensing*. 2022. Vol. 14, № 12. P. 1–21. DOI: https://doi.org/10.3390/rs14122763 (date of access: 20.04.2025).

26. Bošnjak A., Jajac N. Determining priorities in infrastructure management using multicriteria decision analysis. *Sustainability*. 2023. Vol. 15, № 20. P. 1–24. DOI: https://doi.org/10.3390/su152014953 (date of access: 23.04.2025).

27. Prioritizing post-disaster reconstruction projects using an integrated multi-criteria decision-making approach: a case study / Z. Mohammadnazari et al. *Buildings*. 2022. Vol. 12, № 2. P. 136. DOI: https://doi.org/10.3390/buildings12020136 (date of access: 30.04.2025).

28. Boesch G. Computer vision tasks (comprehensive 2025 guide). *viso.ai*. URL: https://viso.ai/deep-learning/computer-vision-tasks/#:~:text=Common%20computer%20vision%20tasks:%20Image,the%20top%20computer%20vision%20tasks (date of access: 08.05.2025).

29. Deep learning for generic object detection: a survey / L. Liu et al. *International journal of computer vision*. 2019. Vol. 128, № 2. P. 261–318. DOI: https://doi.org/10.1007/s11263-019-01247-4 (date of access: 09.05.2025).

30. What is image classification? Basics you need to know. *SuperAnnotate*. URL: https://www.superannotate.com/blog/ image-classification-basics (date of access: 09.05.2025).

31. A comparative performance analysis of classification and segmentation models on bangladeshi pothole dataset. *arXiv.org e-Print archive*. URL: https://arxiv.org/html/2501.06602 (date of access: 09.05.2025).

32. A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision / N. Manakitsa et al. *Technologies*. 2024. Vol. 12, № 2. P. 1–40. DOI: https://doi.org/10.3390/technologies12020015 (date of access: 09.05.2025).

33. Kapoor V. Key differences between object detection and image segmentation. *Training data for your AI projects*. URL: https://mindkosh.com/

blog/key-differences-between-object-detection-and-image-segmentation/ (date of access: 09.05.2025).

34. Удовенко С. Г., Чала Л. Е., Шергін В. Л. Комп'ютерна обробка цифрових статичних зображень з використанням марковських моделей. *Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту*: Материалі международної наукової конференції. Херсон: Видавництво ФЛП Вишемирський В.С., 2018. С. 281–283.

35. Sultana F., Sufian A., Dutta P. Evolution of image segmentation using deep convolutional neural network: a survey. *Knowledge-Based systems*. 2020. Vol. 201-202. P. 106062. DOI: https://doi.org/10.1016 /j.knosys.2020.106062 (date of access: 09.05.2025).

36. Kouidri A. Top instance segmentation models of 2024: a comprehensive guide. *Ikomia. The fastest way to deploy Computer Vision AI*. URL: https://www.ikomia.ai/blog/top-instance-segmentation-models (date of access: 13.05.2025).

37. Ultralytics. Segment. URL: https://docs.ultralytics.com/tasks/segment/ (date of access: 13.05.2025).

38. Top models for instance segmentation reviewed. *Keylabs*. URL: https://keylabs.ai/blog/top-models-for-instance-segmentation-reviewed/ (date of access: 13.05.2025).

39. Xie Y., Chen L. CBLN-YOLO: an improved yolo11n-seg network for cotton topping in fields. *Agronomy*. 2025. Vol. 15, no. 4. P. 996. URL: https://doi.org/10.3390/agronomy15040996 (date of access: 14.05.2025).

40. Rao S. N. YOLOv11 explained: next-level object detection with enhanced speed and accuracy. *Medium*. URL: https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71 (date of access: 14.05.2025).

41. Deep learning-based instance segmentation of cracks from shield tunnel lining images / H. Huang et al. *Structure and infrastructure engineering*. 2020. P. 1–14. DOI: https://doi.org/10.1080/15732479.2020.1838559 (date of access:

16.05.2025).

42. Image data augmentation approaches: a comprehensive survey and future directions / T. Kumar et al. *IEEE access*. 2024. P. 1–38. DOI: https://doi.org/10.1109/access.2024.3470122 (date of access: 17.05.2025).

43. Gallagher J. What is data augmentation? The ultimate guide. *Roboflow Blog*. URL: https://blog.roboflow.com/data-augmentation/ (date of access: 17.05.2025).

44. Rzechowski K. Instance segmentation loss functions | SoftwareMill. *SoftwareMill*. URL: https://softwaremill.com/instance-segmentation-loss-functions/?utm_source=chatgpt.com (date of access: 28.05.2025).

45. Johnson J. M., Khoshgoftaar T. M. Survey on deep learning with class imbalance. *Journal of big data*. 2019. Vol. 6, №. 1. URL: https://doi.org/10.1186/s40537-019-0192-5 (date of access: 28.05.2025).

46. Taherdoost H., Madanchian M. Multi-Criteria decision making (MCDM) methods and concepts. *Encyclopedia*. 2023. Vol. 3, № 1. P. 77–87. URL: https://doi.org/10.3390/encyclopedia3010006 (date of access: 20.05.2025).

47. Madanchian M., Taherdoost H. A comprehensive guide to the TOPSIS method for multi-criteria decision making. *Sustainable social development*. 2023. Vol. 1, № 1. DOI: https://doi.org/10.54517/ssd.v1i1.2220 (date of access: 20.05.2025).

48. Taherdoost H. Decision making using the analytic hierarchy process (AHP); A step by step approach. *International Journal of economics and management system*. 2017. № 2. P. 244-246.

49. Kasperczyk N., Knickel K. The analytic hierarchy process (AHP). VU University. 2006. P. 1 – 6. URL: https://www.iied.org/sites/default/files/pdfs/2022-02/20781G.pdf (дата звернення: 19.05.2025)

50. Casalegno F. Learning to rank: a complete guide to ranking using machine learning. *Medium*. URL: https://medium.com/data-science/learning-to-rank-a-complete-guide-to-ranking-using-machine-learning-4c9688d370d4 (date of access: 20.05.2025).

51. Burges, C. J. C. (2010). *From RankNet to LambdaRank to LambdaMART: An Overview*. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/MSR-TR-2010-82.pdf

52. Пиріг Н. Я., Чала Л. Е. Застосування штучного інтелекту для визначення пріоритетності реконструкції будівель в умовах поствоєнного відновлення. *Сучасні інформаційні системи та технології в цифровому суспільстві*: матеріали Міжнар. наук. конф., (Харків, 10 квіт. – 11 черв. 2025 р). Харків, 2025. С. 61– 62.

53. Crack detection instance segmentation dataset and pre-trained model by asia pacific university. *Roboflow*. URL: https://universe.roboflow.com/asia-pacific-university-a6nbw/crack-detection-lspeh (date of access: 09.04.2025).

54. BD instance segmentation dataset by combined UAV object recognition. *Roboflow*. URL: https://universe.roboflow.com/combined-uav-object-recognition/bd-r7rak (date of access: 13.04.2025).

55. Roboflow: computer vision tools for developers and enterprises. URL: https://roboflow.com/ (date of access: 27.05.2025).

56. Balance classes during YOLO training using a weighted dataloader. *Yasin's Keep*. URL: https://y-t-g.github.io/tutorials/yolo-class-balancing/ (date of access: 27.05.2025).