

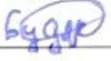
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

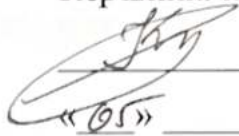
на тему:

**«Інтелектуальна технологія побудови плану будівель по
аерофотозйомці їх дахів»**

Виконав: студент 2 курсу, групи ІСТ-23м
спеціальності 126 – «Інформаційні
системи та технології»

 Владислав БУДЯК

Керівник: к.т.н., доц. каф. САІТ

 Євгеній КРИЖАНОВСЬКИЙ
«05» 12 2024 р.

Опонент: к.т.н., доц. каф. КН

 Володимир ОЗЕРАНСЬКИЙ
«08» 12 2024 р.

Допущено до захисту

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН
«06» 12 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра системного аналізу та інформаційних технологій

Рівень вищої освіти – II-й (магістерський)

Галузь знань – 12 Інформаційні технології

Спеціальність – 126 Інформаційні системи та технології

Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН

«17» 09 2024 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Будяку Владиславу Олександровичу

1. Тема роботи: «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів»,

керівник роботи: Євгеній КРИЖАНОВСЬКИЙ к.т.н., доц. каф. САІТ,

затверджені наказом закладу вищої освіти від «17» 09 2024 року №3/0

2. Строк подання студентом роботи «01» 12 2024 року

3. Вихідні дані до роботи:

- Kaggle Dataset „Aerial Imagery For Roof Segmentation”
<https://www.kaggle.com/datasets/atilol/aerialimageryforroofsegmentation/data>
- Kaggle Dataset „Massachusetts Buildings Dataset”
<https://www.kaggle.com/datasets/balraj98/massachusetts-buildings-dataset>.

4. Зміст текстової частини:

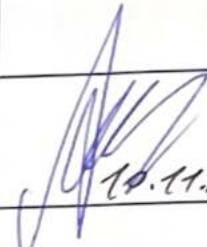
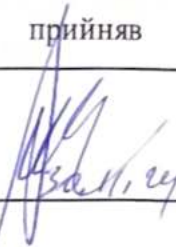
- характеристика поставленої задачі;
- вибір оптимальних налаштувань алгоритму для розв’язання поставленої задачі;
- розроблення інтелектуальної технології побудови плану будівель по аерофотозйомці їх дахів;
- економічна частина.

5. Перелік ілюстративного матеріалу:

- візуалізація даних;

- блок-схема алгоритму сегментації дахів;
- структурні схеми.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Олександр ЛЕСЬКО, к.е.н., проф. каф. ЕПВМ	 10.11.24	 30.11.24

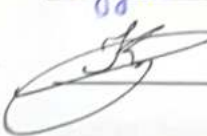
7. Дата видачі завдання « 17 » 09 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	20.09	30.09	виз
2	Вибір оптимальних інформаційних технологій	01.10	15.10	виз
3	Вибір мови програмування та середовища розробки	16.10	31.10	виз
4	Розроблення інтелектуальної технології	01.11	15.11	виз
5	Економічна частина	15.11	25.11	виз
6	Оформлення матеріалів до захисту МКР	25.11	30.11	виз

Студент

Керівник роботи

Владислав БУДЯК

Євгеній КРИЖАНОВСЬКИЙ

АНОТАЦІЯ

УДК 004.09

Будяк В.О. Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2024. 110 с.

На укр. мові. Бібліогр.: 36 назв; рис.: 35; табл.: 12.

У магістерській кваліфікаційній роботі звернено увагу на проблематику автоматизації побудови планів будівель, яка є актуальною у сфері містобудування та інфраструктурного планування. Досліджено методи та підходи до обробки аерофотознімків, що дозволяють визначати контури дахів будівель з високою точністю та оптимізувати процеси планування території.

Запропоновані технології глибинного навчання та сегментації спрямовані на вирішення завдань аналізу аерофотознімків шляхом автоматизованого виявлення будівельних об'єктів та покращення точності даних для міського планування. Результати досліджень можуть бути використані для розробки сучасних інструментів обробки зображень, які сприятимуть ефективному управлінню міськими ресурсами. Об'єкт досліджень – процес створення інтелектуальної технології для аналізу та сегментації дахів будівель на основі аерофотознімків. Галузь застосування – системи автоматизованого проектування, геоінформаційні системи та інструменти містобудівного планування.

Ілюстративна частина складається з 7 плакатів.

Ключові слова: глибинне навчання, інтелектуальні технології, сегментація, аерофотозйомка, міське планування, машинне навчання, дахи будівель.

ABSTRACT

Budiak V.O. Intelligent Technology for Building Plans Creation Based on Aerial Roof Imagery. Master's Qualification Thesis in the specialty 126 – Information Systems and Technologies, Educational and Professional Program – Information Technologies for Data and Image Analysis. Vinnytsia: VNTU, 2024. 110 pages.

In Ukrainian. Bibliography: 36 references; figures: 35; tables: 12.

The master's qualification thesis addresses the problem of automating the creation of building plans, which is highly relevant in the field of urban development and infrastructure planning. The study examines methods and approaches for processing aerial imagery to accurately identify building roof contours and optimize urban planning processes.

The proposed deep learning and segmentation technologies aim to solve the challenges of analyzing aerial imagery by automating the detection of building objects and improving data accuracy for urban planning. The research findings can be applied to develop modern image processing tools that facilitate efficient urban resource management.

Research object: the process of creating an intelligent technology for analyzing and segmenting building roofs based on aerial imagery.

Application domain: computer-aided design systems, geographic information systems, and urban planning tools.

The illustrative part comprises 7 posters.

Keywords: deep learning, intelligent technologies, segmentation, aerial imagery, urban planning, machine learning, building roofs.

ЗМІСТ

ВСТУП.....	4
1 ХАРАКТЕРИСТИКА ПОСТАВЛЕНОЇ ЗАДАЧІ	6
1.1 Актуальність побудови плану будівель по аерофотозйомці їх дахів	6
1.2 Постановка задачі та вихідні передумови	12
1.3 Отримання та підготовка даних.....	15
1.4 Висновки	19
2 ВИБІР ОПТИМАЛЬНИХ НАЛАШТУВАНЬ АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	20
2.1 Попередній аналіз даних	20
2.2 Вибір мови програмування та середовища розробки.....	23
2.3 Вибір оптимальних налаштувань алгоритму для сегментації дахів на аерофотознімках.....	28
2.4 Висновки	37
3 РОЗРОБЛЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ТЕХНОЛОГІЇ ПОБУДОВИ ПЛАНУ БУДІВЕЛЬ ПО АЕРОФОТОЗЙОМЦІ ЇХ ДАХІВ	39
3.1 Алгоритм вирішення задачі	39
3.2 Підготовка до роботи.....	44
3.3 Підготовка результатів алгоритму ідентифікації будівель на зображенні	56
3.4 Детальний аналіз результатів навчання моделі машинного навчання	61
3.5 Порівняння точності запропонованої технології з відомими підходами...	64
3.6 Висновки	66
4 ЕКОНОМІЧНА ЧАСТИНА.....	68
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	68
4.2 Розрахунок узагальненого коефіцієнта якості розробки	69
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	71
4.3.1 Витрати на оплату праці.....	71
4.3.2 Відрахування на соціальні заходи.....	74
4.3.3 Сировина та матеріали	74

	3
4.3.4 Розрахунок витрат на комплектуючі	76
4.3.5 Спецустаткування для наукових (експериментальних) робіт	76
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт	77
4.3.7 Амортизація обладнання, програмних засобів та приміщень	78
4.3.8 Паливо та енергія для науково-виробничих цілей	80
4.3.9 Службові відрядження	81
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації	81
4.3.11 Інші витрати.....	81
4.3.12 Накладні (загальновиробничі) витрати	82
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором	83
4.5 Висновки	87
ВИСНОВКИ.....	89
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	91
Додаток А (обов'язковий). Технічне завдання.....	95
Додаток Б (обов'язковий). Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	98
Додаток В (довідковий). Лістинг програми	99
Додаток Г (обов'язковий). Ілюстративна частина	107

ВСТУП

Актуальність дослідження побудови плану будівель за даними аерофотозйомки обумовлена потребою у точних інструментах для оцінки стану будівель та планування нових будівельних проектів. Використання сучасних інформаційних технологій у цій сфері дозволить значно підвищити точність і швидкість аналізу, мінімізувати ризики та забезпечити ефективне управління інфраструктурою. Це робить розробку нових методів розпізнавання зображень не тільки актуальною, але й необхідною для подальшого розвитку урбаністики та будівельної галузі.

Мета і завдання роботи. Метою дослідження є підвищення точності та ефективності побудови планів будівель за даними аерофотозйомки.

Для досягнення поставленої мети потрібно виконати такі завдання:

- Провести аналіз актуальності та необхідності побудови плану будівель за даними аерофотозйомки.
- Здійснити вибір оптимальних технологій та моделей для обробки аерофотознімків.
- Розробити технологію побудови плану будівель за даними аерофотозйомки.
- Провести порівняння розробленої інтелектуальної технології з існуючими моделями.

Об'єктом дослідження магістерської кваліфікаційної роботи є процес створення плану будівель за даними аерофотозйомки.

Предметом дослідження магістерської кваліфікаційної роботи є методи інформаційних технологій, що застосовуються для аналізу та обробки аерофотознімків.

Новизна одержаних результатів. Подальшого розвитку набула технологія створення планів будівель, яка, на відміну від існуючих, використовує моделі DETR (DEtection TRansformer) та Vision Transformers (ViTs) для автоматичної ідентифікації та сегментації дахів будівель. Це забезпечує високу точність побудови планів, навіть для складних конфігурацій

дахів.

Практичне значення. Результати магістерської кваліфікаційної роботи можуть бути застосовані для автоматизації процесу створення планів будівель, що є важливим для міського планування, реконструкції об'єктів та управління територіями. Використання розробленої технології дозволить значно скоротити час і ресурси, необхідні для підготовки архітектурних планів.

Апробація результатів магістерської кваліфікаційної роботи. За результатами роботи подано до друку тези на конференцію «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)».

Публікації результатів магістерської кваліфікаційної роботи. Опубліковано статтю у фаховому виданні [1] та подано до друку тези на конференцію «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.) [2].

1 ХАРАКТЕРИСТИКА ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1 Актуальність побудови плану будівель по аерофотозйомці їх дахів

Аерофотозйомки місцевості це роботи, основи яких створюють ортофотоплани і фотосхеми місцевості. Аерофотозйомка ділянки дозволяє робити зйомку в певному місці на потрібній висоті і в потрібний момент часу. Якісна аерофотозйомка земельної ділянки відрізняється високою точністю і має безліч переваг [3].

У сучасному світі точне картографування та планування міських територій набувають все більшого значення. Одним із найефективніших методів досягнення цієї мети є аерофотозйомка дахів будівель, яка має кілька суттєвих переваг, що визначають її актуальність.

По-перше, аерофотозйомка дозволяє отримати зображення з високою точністю та деталізацією. Це особливо важливо для створення детальних планів будівель, де кожна деталь може мати велике значення, наприклад, для інженерних комунікацій, архітектурних особливостей та безпеки. Точні дані, отримані за допомогою аерофотозйомки, дозволяють не лише створювати актуальні карти, але й оновлювати їх у відповідь на зміни у міському середовищі. Сучасні методи обробки зображень, такі як фотограмметрія, забезпечують високу якість отриманих даних, що дозволяє використовувати їх для різних цілей, включаючи проєктування та будівництво нових об'єктів. Наприклад, книга "Introduction to Modern Photogrammetry" авторів Едуарда Маліновського та Дмитра Погорілого надає комплексне розуміння основ фотограмметрії та її застосування, включаючи аерофотозйомку.

По-друге, аерофотозйомка є надзвичайно ефективним та швидким методом збору даних. У порівнянні з традиційними наземними методами, зйомка з повітря значно прискорює процес картографування. Це дозволяє швидко отримувати та аналізувати дані, що особливо важливо в умовах динамічного розвитку міських територій. Завдяки цьому, можна оперативно реагувати на

зміни та забезпечувати актуальність картографічних матеріалів. Використання дронів для аерофотозйомки дозволяє виконувати зйомку великих територій за короткий час, що знижує витрати та підвищує ефективність процесу. Наприклад, сайт "National Center for Airborne Laser Mapping (NCALM)" містить багато інформації про сучасні технології аерофотозйомки та їх застосування у картографуванні.

Крім того, аерофотозйомка забезпечує безпеку та доступність при дослідженні важкодоступних або небезпечних об'єктів. Використання дронів та інших засобів аерофотозйомки дозволяє уникати небезпеки для дослідників, знижуючи ризики під час роботи з аварійними або старими спорудами, а також об'єктами, розташованими у важкодоступних районах. Це особливо актуально для інспекції інфраструктурних об'єктів, таких як мости, дамби, високовольтні лінії тощо. Завдяки цій технології можна отримати точні дані без необхідності фізичної присутності людини на небезпечних або важкодоступних об'єктах [3].

Аерофотозйомка є також економічно доцільним методом. Використання дронів та іншого аерофотозйомочного обладнання знижує потребу у великій кількості персоналу та дорогому обладнанні, що робить цей метод більш економічно вигідним порівняно з традиційними методами. Економія коштів дозволяє спрямовувати ресурси на інші важливі проєкти та задачі, підвищуючи загальну ефективність роботи. Це особливо важливо для муніципалітетів та приватних компаній, які прагнуть оптимізувати свої витрати.

Сучасні технології дозволяють автоматизувати обробку та аналіз даних, отриманих за допомогою аерофотозйомки, інтегруючи їх у геоінформаційні системи (ГІС). Це забезпечує зручний доступ до інформації, спрощує процес прийняття рішень та підвищує точність планування. ГІС-технології дозволяють створювати інтерактивні карти та моделі, що використовуються для аналізу та прогнозування розвитку міських територій. Наприклад, на основі аерофотозйомки можна створювати тривимірні моделі будівель, що допомагає візуалізувати та планувати нові забудови.

Нарешті, аерофотозйомка має значний екологічний аспект. Завдяки точним даним про розташування та стан дахів будівель можна ефективніше

планувати заходи з енергозбереження та встановлення екологічно чистих технологій, таких як сонячні панелі чи зелені дахи. Це сприяє підвищенню енергоефективності та зниженню негативного впливу на навколишнє середовище. Наприклад, аналіз даних з аерофотозйомки дозволяє визначити оптимальні місця для встановлення сонячних панелей, що підвищує ефективність їх роботи та зменшує витрати на енергозабезпечення.

Таким чином, використання аерофотозйомки для побудови плану будівель є сучасним, ефективним та економічно обґрунтованим методом, який не лише полегшує процес планування та управління міським середовищем, але й сприяє підвищенню якості життя мешканців, безпеки та екологічної стійкості. Це робить аерофотозйомку незамінним інструментом у сфері урбаністики та архітектурного планування.

Містобудування — комплексна багатогранна діяльність суспільства, спрямована на створення матеріально-просторового середовища життєдіяльності людини в поселеннях та районах розселення.

Галузь науки і техніки, яка веде дослідження інженерно-технічних, соціально-економічних та екологічних проблем формування життєвого середовища, що включає конструювання систем населених місць, їх планування й забудови.

В межах спеціальності проводиться: розроблення методології аналізу та синтезу різноманітних типів містобудівних об'єктів (регіонів, міст, сіл та їх структурних елементів), створення теоретичних підстав сучасного містобудування, обґрунтування державної містобудівної політики, інформаційного забезпечення містобудівної діяльності.

Результати досліджень мають сприяти забезпеченню процесів управління розвитком населених місць та територій, планування забудови, реконструкції та експлуатації населених місць та регіонів відповідно до потреб населення й виробництва [4].

Важливою проблемою містобудівної справи є оперативне створення детального плану місцевості за даними аерофотозйомки. Це завдання є критично важливим для ефективного планування міської інфраструктури, розвитку

територій, а також для організації будівельних робіт. Використання аерофотозйомки дозволяє отримати високоточні зображення великої площі в найкоротші терміни, що є значною перевагою у порівнянні з традиційними методами картографування. Проте, незважаючи на всі переваги, ця технологія має і свої складнощі та обмеження.

Однією з найбільш складних задач в оперативному створенні детального плану місцевості є відрисовування фундаменту будівлі по її даху. Це завдання ускладнюється тим, що сам фундамент не видно на аерофотознімках, і можна лише здогадуватись про його розташування на основі зйомки з різних ракурсів. Тобто, на знімках видно тільки дах будівлі, що значно ускладнює процес визначення точних контурів фундаменту.

Для точного відрисовування фундаменту необхідно враховувати різні фактори. По-перше, необхідно мати високоякісні зображення, зроблені з різних кутів та висот. Це дозволяє отримати максимально точну інформацію про геометрію будівлі. По-друге, важливо враховувати тіні, що можуть спотворювати зображення та створювати хибні враження про форму та розміри будівлі. Третім важливим аспектом є врахування перспективних спотворень, які виникають через те, що зйомка проводиться з різних висот і кутів.

Сучасні технології дозволяють частково автоматизувати процес відрисовування фундаментів за допомогою спеціалізованого програмного забезпечення, яке може аналізувати зображення та визначати ймовірні контури фундаменту на основі геометрії даху. Проте, навіть при використанні таких технологій, людський фактор залишається важливим, оскільки лише досвідчений фахівець може зробити остаточні висновки на основі аналізу отриманих даних [4].

Також, варто зазначити, що для покращення точності визначення контурів фундаменту можуть використовуватись додаткові методи зйомки, такі як лазерне сканування (LIDAR), яке дозволяє отримувати тривимірні моделі місцевості з високою точністю. Інтеграція даних з різних джерел може значно підвищити якість та точність кінцевого результату.

З іншого боку, розвиток алгоритмів машинного навчання і штучного інтелекту відкриває нові перспективи для автоматизації процесу аналізу аерофотознімків. Такі алгоритми можуть навчатися на великій кількості даних і самостійно визначати характерні ознаки, що дозволяють робити висновки про розташування фундаментів. Однак, впровадження таких технологій потребує значних обчислювальних ресурсів та наявності великого обсягу навчальних даних.

У підсумку, процес відрисовування фундаменту будівлі по її даху на основі аерофотозйомки є складним і багатогранним завданням, яке потребує використання сучасних технологій, досвіду фахівців та інтеграції різноманітних методів зйомки і аналізу даних. Незважаючи на існуючі виклики, цей напрямок має великий потенціал для розвитку і вдосконалення, що в перспективі дозволить значно підвищити ефективність і точність містобудівної справи.

Щомісячно будується величезна кількість нових будинків, і темпи цього будівництва постійно зростають у відповідь на зростаючий попит на житлову та комерційну нерухомість. Це створює значний виклик для міських планувальників і будівельних інспекторів, які повинні контролювати всі нові об'єкти, забезпечуючи їх відповідність будівельним нормам і стандартам. Масштаби сучасного будівництва ускладнюють процес моніторингу та документування всіх новобудов вручну. Традиційні методи інспекції стають недостатніми, оскільки вони вимагають багато часу та ресурсів, щоб охопити всі нові будівельні майданчики [4].

В умовах стрімкого зростання обсягів будівництва виникає потреба в більш ефективних і швидких методах моніторингу та документування. Аерофотозйомка є одним з найсучасніших і найефективніших інструментів для вирішення цієї проблеми. Вона дозволяє швидко отримувати високоякісні зображення великих територій, що значно прискорює процес збору даних про нові будівлі.

Оперативність аерофотозйомки є її однією з ключових переваг. За допомогою дронів та інших засобів аерофотозйомки можна за короткий час охопити великі площі та отримати актуальні дані про стан будівництва. Це

особливо важливо в умовах міського середовища, де нові будівельні майданчики з'являються щомісяця, і необхідно вчасно відстежувати їхній стан.

Точність і детальність отриманих зображень дозволяють фіксувати розташування будівель, їхню геометрію та стан на кожному етапі будівництва. Це дає змогу будівельним інспекторам і міським планувальникам мати точне уявлення про те, як просуваються роботи, і вчасно виявляти будь-які відхилення від плану. Завдяки цьому можна вчасно вживати необхідних заходів для корекції та забезпечення відповідності будівельним нормам.

Аерофотозйомка також є важливим інструментом для документування процесу будівництва. Зображення, отримані під час аерофотозйомки, можуть використовуватися для складання звітів, аналізу виконаних робіт і підтвердження відповідності будівельним нормам. Це спрощує процес підготовки документації та підвищує прозорість будівельних проектів.

Крім того, використання аерофотозйомки дозволяє значно знизити витрати на проведення інспекцій і обстежень будівельних майданчиків. Традиційні методи часто вимагають значних ресурсів, тоді як аерофотозйомка може виконуватися швидко і ефективно, зменшуючи потребу в залученні великої кількості персоналу. Це підвищує ефективність роботи будівельних інспекторів і міських планувальників, дозволяючи їм зосередитися на найважливіших аспектах контролю та управління будівельними проектами.

Важливість аерофотозйомки для сучасного містобудування важко переоцінити. Завдяки своїм перевагам вона дозволяє оперативно отримувати точні та актуальні дані про міські території, що є критично важливим для ефективного планування та управління міським середовищем. Використання аерофотозйомки сприяє підвищенню якості та швидкості прийняття рішень у будівельних проектах, що в кінцевому рахунку сприяє сталому розвитку міст і покращенню якості життя їхніх мешканців. Темпи введення житла в експлуатацію зображено на рисунку 1.1 [5].



Рисунок 1.1 – Темпи введення житла в експлуатацію

1.2 Постановка задачі та вихідні передумови

Однією з ключових проблем, з якою стикаються фахівці в галузі містобудування, є оперативна та точна розробка детальних планів територій на основі даних аерофотозйомки. Цей процес є важливим для забезпечення точного проектування та раціонального планування забудови міських територій. Залучення аерофотознімків дає можливість отримати об'єктивні та детальні дані про стан території, що є важливим для планування інфраструктури, ефективного використання земельних ресурсів та збереження екологічного балансу [6].

Якісні детальні плани на основі аерофотознімків забезпечують оптимізацію використання наявних земельних ресурсів і грають важливу роль у вирішенні можливих конфліктів між різними зацікавленими сторонами, такими як забудовники, органи місцевого самоврядування та місцеві жителі. Такі плани сприяють уникненню протиріч та забезпеченню взаєморозуміння під час ухвалення рішень про забудову, розвиток інфраструктури та розширення

міських територій. Окрім того, детальні плани сприяють забезпеченню правомірного та прозорого використання земель, що має велике значення в умовах сучасної урбанізації.

Одним із найбільш складних викликів, що виникають у контексті аналізу аерофотознімків для створення детальних планів, є завдання реконструкції контурів фундаментів будівель. Це завдання стає особливо складним, коли доступна інформація обмежується лише зображеннями дахів будівель, оскільки самі фундаменти часто залишаються поза видимістю. Спеціалісти змушені використовувати різні методи аналізу зображень і комбінувати дані, отримані з різних ракурсів, для побудови максимально точних моделей. Цей процес потребує врахування численних факторів, серед яких є форма даху, архітектурні особливості будівлі, матеріали, що використовувалися для її будівництва, а також геометричні співвідношення між елементами конструкції.

Окрім того, під час аналізу аерофотознімків фахівці стикаються з численними технічними проблемами, які можуть вплинути на якість отриманих даних. Серед таких проблем — перспективні спотворення, зміни освітлення, погодні умови та інші зовнішні фактори, що можуть викликати погіршення якості зображень. Ці труднощі вимагають залучення сучасних технологій, зокрема систем геоінформаційного моделювання, які дозволяють не лише коригувати спотворення, а й проводити детальний аналіз отриманих даних. Це допомагає адаптувати результати до реальних умов на місцевості та зменшити похибки під час проектування [6].

На сьогоднішній день фахівці можуть використовувати сучасні програмні рішення, які поєднують алгоритми машинного навчання та комп'ютерної графіки, для підвищення точності моделювання і прогнозування. Це дозволяє не тільки точно відтворювати контури фундаменту будівлі, а й проводити оцінку можливих впливів забудови на навколишнє середовище, соціальні умови та економічні показники розвитку регіону. Впровадження таких технологій є важливим етапом у забезпеченні сталого розвитку міст і населених пунктів, оскільки воно сприяє збереженню балансу між розвитком інфраструктури та захистом екологічних інтересів.

Сучасні системи геоінформаційного моделювання, такі як ArcGIS, QGIS, та інші програмні пакети, відкривають нові можливості для обробки даних аерофотозйомки та побудови точних планів територій. Окрім цих програмних рішень, популярними є також інструменти для обробки зображень, такі як Adobe Photoshop та Digital. Завдяки застосуванню різних технологій, включаючи методи аналізу кольорів та сегментації зображень [6-8], стає можливим точніше відтворювати контури будівель і фундаментів. Наприклад, один з ефективних підходів полягає у порівнянні нових аерофотознімків із раніше створеними планами тієї ж території [9]. Це дозволяє уточнити розміри й форму будівель і їхніх фундаментів навіть у випадках, коли безпосередньо отримана інформація обмежена лише дахами будівель.

Проте, традиційні методи аналізу аерофотознімків мають свої обмеження. Одним з основних недоліків є їхня низька точність під час роботи з дахами складної конфігурації, особливо якщо місцевість має складний рельєф. Іншими обмежуючими факторами є насиченість зображення великою кількістю об'єктів, які можуть бути візуально схожими на дахи будівель, та варіативність кольорів дахів, які не завжди контрастно виділяються на тлі інших елементів ландшафту. Крім того, наявність об'єктів із подібними формами може значно ускладнити точне розпізнавання будівель і, відповідно, їх фундаментів.

Зважаючи на вищезазначені виклики, сучасні інтелектуальні методи розпізнавання зображень на основі штучного інтелекту набувають все більшого значення у сфері містобудування. Інтелектуальні системи, зокрема на базі трансформерів ViTs та псевдоміток [10-15], демонструють значні успіхи у розв'язанні проблем розпізнавання дахів та їх точного аналізу. Використання машинного навчання для навчання моделей на великих масивах даних дозволяє значно підвищити точність ідентифікації контурів будівель, зокрема їх фундаментів. Це стає можливим завдяки автоматичній корекції спотворень та адаптації алгоритмів до конкретних умов місцевості, що дозволяє мінімізувати вплив зовнішніх чинників, таких як погодні умови чи низька якість зображень.

Тому дослідження, спрямовані на підвищення точності ідентифікації фундаментів будівель за даними аерофотозйомки, мають велике значення для

подальшого розвитку містобудування. Метою такого дослідження є не лише покращення точності ідентифікації будівель, але й забезпечення ефективного використання ресурсів, збереження екологічної рівноваги та забезпечення сталого розвитку урбаністичних територій. Використання сучасних інтелектуальних технологій розпізнавання зображень є важливим кроком на шляху до підвищення якості проектування міських територій та створення більш стійких і функціональних інфраструктурних рішень.

1.3 Отримання та підготовка даних

Для ефективного вирішення задачі визначення контурів фундаменту будівель на основі аерофотозйомки доцільно використовувати датасет "Aerial Imagery for Roof Segmentation" [16], розміщений на платформі Kaggle. Цей датасет був створений спеціально для задач, пов'язаних із сегментацією дахів будівель на аерофотознімках, що робить його одним із найцінніших інструментів для досліджень у сфері комп'ютерного зору. Використання такого набору даних забезпечує доступ до високоякісних і добре розмічених зображень, які дозволяють навчати складні моделі глибокого навчання для точного виявлення та ідентифікації дахів будівель на зображеннях з висоти. Завдяки цьому досягається значне підвищення точності розроблених алгоритмів, що допомагає в автоматизації процесів аналізу великих масивів даних і використовується для вирішення задач у сфері міського планування та управління територіями.

Набір даних "Aerial Imagery for Roof Segmentation" є незамінним ресурсом для вирішення завдань сегментації дахів, які є критичним елементом при створенні детальних топографічних карт і планів міст. Використання цих карт і планів дозволяє містам приймати інформовані рішення щодо майбутнього розвитку інфраструктури, проектування нових житлових або промислових зон, а також ефективного використання земельних ресурсів. У зв'язку з швидким темпом урбанізації та постійним зростанням обсягу даних, отриманих з аерофотозйомок, автоматизація аналізу цих даних є життєво важливою для підвищення ефективності процесів планування та розвитку міських просторів.

Впровадження передових методів глибокого навчання, таких як згорткові нейронні мережі (CNN), дозволяє суттєво підвищити швидкість і точність обробки аерофотознімків, що в свою чергу покращує якість прийняття рішень щодо управління міськими ресурсами [16].

Цей датасет містить високоточні аерофотознімки дахів будівель, а також відповідні бінарні маски, які чітко позначають контури дахів і відокремлюють їх від навколишнього середовища. Це дозволяє моделі в процесі навчання отримувати точні уявлення про форми дахів і їх характеристики, що надалі дає можливість правильно ідентифікувати будівлі на нових знімках. Ілюстрація основних типів дахів, що зустрічаються на аерофотознімках, наведена на рисунку 1.2.



Рисунок 1.2 – Ілюстрація типів дахів

Датасет розділений на три основні частини: тренувальні дані (train), дані для валідації (val) та тестові дані (test). Кожен з цих розділів відіграє важливу

роль у процесі навчання і тестування моделі, допомагаючи на кожному етапі оцінювати її продуктивність і точність [16].

Тренувальний набір є найбільшим і найважливішим розділом датасету. Він включає значну кількість зображень разом із відповідними масками, що дозволяє моделям навчитися розпізнавати та класифікувати дахи будівель. Моделі, треновані на великому обсязі таких даних, отримують здатність ефективно визначати різні типи дахів і виводити закономірності, що забезпечує високу точність при сегментації. Валідуючий набір використовується для перевірки, наскільки добре модель узагальнює отримані знання і може застосовувати їх до нових даних. Це важливий етап у процесі навчання, який допомагає виявляти проблеми перенавчання і гарантувати, що модель здатна працювати з різними типами зображень, не адаптуючись лише до конкретних випадків. Тестовий набір, який складається з нових зображень і масок, не використаних на етапах тренування і валідації, дає можливість об'єктивно оцінити остаточну ефективність моделі на абсолютно нових даних.

Технічні характеристики датасету мають велике значення для його застосування в задачах сегментації дахів. Завдяки високій роздільній здатності зображень можна з великою точністю ідентифікувати об'єкти навіть на найдетальніших аерофотознімках, що є особливо важливим для завдань, які вимагають високої точності. Крім того, всі зображення мають однакові розміри, що спрощує їх обробку перед використанням у моделях глибинного навчання. Уніфіковані розміри дозволяють уникнути додаткових етапів попередньої обробки, таких як нормалізація чи зміна масштабу, що значно полегшує процес підготовки даних для моделювання.

Особливо важливою перевагою цього датасету є висока якість розмітки. Маски сегментації представлені у бінарному вигляді, що чітко розділяє дахи будівель від фону і дає можливість моделі точно вчитися розпізнавати ці об'єкти на нових зображеннях. Якість розмітки є критичною, оскільки навіть найменші похибки у цьому процесі можуть значно вплинути на результат роботи моделі. Точна розмітка гарантує, що моделі зможуть навчитися правильно виділяти будівлі на основі точних даних, що забезпечує високу точність сегментації. Крім

того, датасет є масштабованим і може бути застосований до задач, які потребують обробки великих обсягів аерофотознімків. Це робить його ідеальним вибором для вирішення реальних практичних завдань, таких як моніторинг міської інфраструктури або контроль за забудовою територій.

Незважаючи на те, що цей датасет "Aerial Imagery for Roof Segmentation" [15] був створений для вирішення конкретної задачі сегментації дахів, його потенційне застосування є значно ширшим. Він може бути використаний для інших завдань комп'ютерного зору, таких як класифікація об'єктів на аерофотознімках, моніторинг міських забудов, оцінка стану будівель тощо. Це робить датасет універсальним ресурсом для виконання широкого спектра задач, пов'язаних із аналізом міської інфраструктури та управління міськими просторами. У сучасних умовах, коли автоматизація аналізу аерофотознімків стає все більш актуальною, використання таких інструментів дозволяє значно скоротити час і зусилля, необхідні для обробки великих обсягів даних.

Цей датасет також є чудовою базою для навчання глибоких нейронних мереж, таких як згорткові нейронні мережі (CNN), які мають широке застосування у задачах обробки зображень. Високоякісні дані, надані у датасеті, дозволяють моделі вчитися на прикладах реальних аерофотознімків і досягати високих результатів при вирішенні практичних завдань. Завдяки цьому такі моделі можуть бути застосовані для аналізу міської інфраструктури, автоматизованого планування територій, оцінки стану будівель, а також для створення детальних карт забудованих територій.

Загалом, датасет "Aerial Imagery for Roof Segmentation" [16] є одним із найбільш корисних і потужних інструментів для вирішення задач, пов'язаних із сегментацією дахів на аерофотознімках. Він відкриває великі можливості для розробки високоточних моделей глибокого навчання, які можуть бути застосовані для різних галузей, включаючи міське планування, аналіз інфраструктури, моніторинг міських забудов і багато інших сфер, де необхідна автоматизація процесів обробки супутникових і аерофотознімків.

1.4 Висновки

В умовах швидкої урбанізації та інтенсивного будівництва актуальність створення детальних планів місцевості за допомогою аерофотозйомки дахів будівель є надзвичайно високою. Ця технологія дозволяє отримувати точні зображення, необхідні для ефективного планування та управління міською інфраструктурою.

Основним завданням є визначення місцезнаходження фундаментів будівель на основі знімків дахів, що потребує сучасних методів глибинного навчання для точного виявлення контурів будівель. Для цього використано датасет «Aerial Imagery for Roof Segmentation» із Kaggle, який забезпечує високоякісні аерофотознімки та відповідні маски сегментації.

Огляд ГІС-пакетів, таких як ArcGIS і QGIS, показав їхню важливість у просторовому аналізі. ArcGIS надає потужні інструменти для моделювання територій, тоді як QGIS є доступною платформою з відкритим кодом. Проте традиційні методи в цих системах мають обмеження в умовах складного рельєфу чи густої забудови.

Сучасні моделі, такі як DeTr (Detection Transformer) і трансформери (ViTs), демонструють високу ефективність у задачах сегментації завдяки здатності обробляти складні сцени та визначати об'єкти навіть у густій забудові чи за наявності перешкод. Використання цих підходів дозволяє підвищити точність сегментації та автоматизувати процес аналізу даних, що є ключовим для створення ефективних інструментів планування та розвитку міських територій.

2 ВИБІР ОПТИМАЛЬНИХ НАЛАШТУВАНЬ АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Попередній аналіз даних

Таблиця містить класифікацію основних типів дахів, які найчастіше зустрічаються в містобудівній діяльності. Ця класифікація враховує конструктивні особливості, функціональне призначення та поширеність дахів у різних архітектурних стилях і типах будівель. Поділ на типи дахів дозволяє більш точно оцінювати їх роль у формуванні міської інфраструктури та плануванні забудови.

Наприклад, двосхилі дахи є найбільш поширеними у житлових районах завдяки простоті конструкції та ефективності водовідведення. Вальмові та шатрові дахи використовуються для споруд, які потребують кращого захисту від кліматичних умов, таких як сильний вітер чи снігові навантаження. Плоскі дахи, зокрема, популярні у багатоповерхових і промислових будівлях, оскільки забезпечують можливість організації додаткових функціональних зон, таких як тераси чи сонячні панелі.

Класифікація дахів також враховує особливості їх геометричних форм і матеріалів, що визначають методи будівництва і подальшого обслуговування. Такий підхід дозволяє ефективніше враховувати специфіку регіональних кліматичних умов [17].

Ця класифікація важлива для обробки аерофотознімків, оскільки кожен тип даху має характерні геометричні риси, які враховуються при розробці алгоритмів сегментації. Використання адаптованих методів аналізу зображень для різних видів дахів дозволяє досягти більшої точності при виявленні та відокремленні об'єктів на аерофотознімках.

Такий підхід є ключовим для автоматизації процесів побудови планів будівель, забезпечуючи точність, ефективність та відповідність проєктів реальним потребам урбаністичного середовища.

У таблиці 2.1 наведено основні види дахів, які виділяють у містобудівній діяльності [17].

Таблиця 2.1 - Основні види дахів

Вид даху	Характеристика
Односхилий дах	Опирається своєю несучою конструкцією на зовнішні стіни, що знаходяться на різних рівнях.
Двосхилий дах	Є найпоширенішою класичною конструкцією. До численних варіантів даного типу треба віднести дахи з рівномірним або нерівномірним кутом нахилу ската або ж розміром карнизного звису.
Вальмовий дах	Чотирискатний: два ската представляють собою трапецію, а два інших, з боку торцевих стін - трикутники (вони називаються “вальмами”).
Шатровий дах	Всі скати такого даху - у вигляді рівнобедрених трикутників, які сходяться в одній точці. Визначальним елементом є симетричність. Застосовується для будівель у формі квадрата або рівностороннього багатокутника.
Мансардний дах	Для збільшення обсягу житлового горищного приміщення (мансарди) часто виконуються скати різних нахилів: нижні - крутіші і верхні - більш пологі.
Плоскі дахи	Це - такий дах, в якого контури стін збігаються з контуром даху. На відміну від скатних дахів, на плоских дахах не застосовують, як покрівельні, штучні і листові матеріали.

Більш розширені класифікації види дахів, розширена класифікація видів дахів наведена на рисунку 2.1 та найпоширеніші види дахів наведена на рисунку 2.2.

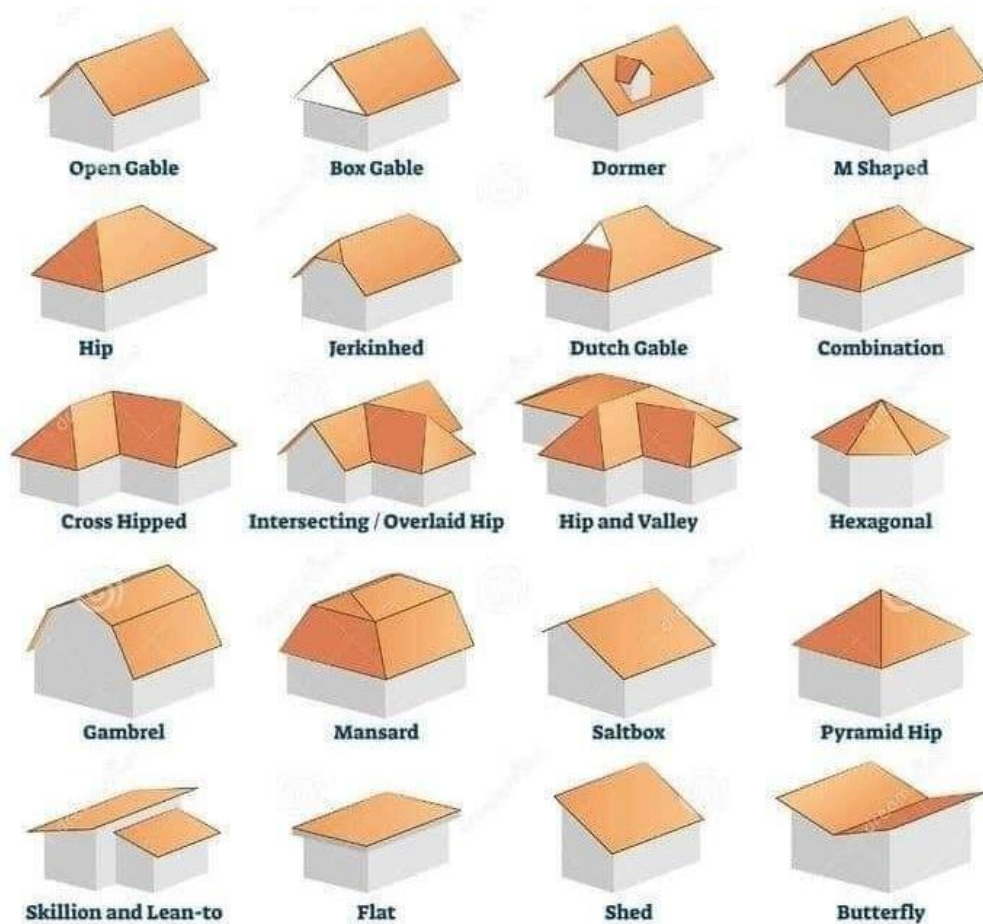


Рисунок 2.1 – Розширена класифікація видів дахів [18]

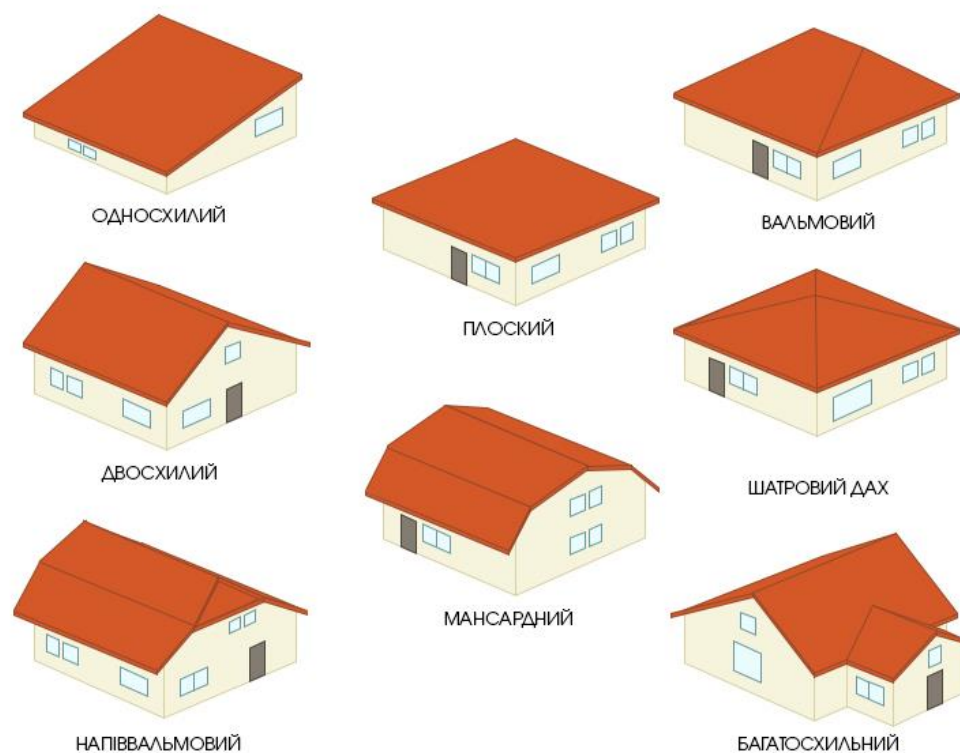


Рисунок 2.2 – Найпоширеніші види дахів [19]

У більшості випадків зустрічаються класи з односхилими ("shed"), двосхилими ("open/box gable"), плоскими ("flat") та вальмовими ("hip") дахами. Тому, у першому наближенні, можна звужити постановку задачі до ідентифікації координат фундаментів саме цих чотирьох видів дахів. Для їх ідентифікації пропонуємо наступний алгоритм: спочатку вхідне зображення передається в модель детекції груп дахів на базі DETR ("DEtection TRansformer") для кращої локалізації. Результатами моделі детекції є регіони груп заданих дахів, які передаються в модель сегментації дахів на базі ViTs. Отримані маски сегментації з моделі на базі ViTs поєднуються з парами відповідних оригінальних зображень, в результаті чого отримуються зображення, де з точністю до пікселів визначено дахи. До отриманих зображень застосовується комплекс методів післяоброблення для знаходження контурів даху та ліній гребеня. Для односхилих ("shed") та плоских ("flat") дахів достатньо знайти контури дахів, оскільки в даного типу дахів відсутня лінія гребеня. Це включає алгоритми пошуку контурів Canny Edge Detection та методи фільтрації контурів. Після цього знаходяться та відрисовуються лінії на зображенні на основі результатів післяоброблення.

2.2 Вибір мови програмування та середовища розробки

У процесі виконання дипломного проєкту було обрано мову програмування Python та платформу Kaggle як основні інструменти для розробки та дослідження. Python - це високорівнева мова програмування, яку створив Гвідо ван Россум і вперше випустив 1991 року. Історія Python сягає корінням у кінець 1980-х років, коли ван Россум почав роботу над нею як побічний проєкт у Centrum Wiskunde & Informatica в Нідерландах [20]. Kaggle — платформа для змагань з аналітики та передбачувального моделювання, в рамках якого статистики та добувачі даних конкурують у створенні найкращих моделі для прогнозування та опису даних, запропонованих компаніями або користувачами. Цей краудсорсинговий підхід ґрунтується на тому, що є безліч стратегій, які можуть бути застосовані до будь-якого завдання з передбачувального

моделювання, і наперед не відомо, яка методика або аналітичний підхід буде найбільш ефективним [21]. Такий вибір пояснюється численними перевагами, які ці інструменти надають для задач обробки даних, машинного навчання та, зокрема, роботи з зображеннями. Python, завдяки своїй простоті та багатій екосистемі бібліотек, забезпечує ефективне рішення для складних алгоритмічних завдань, а Kaggle надає зручне середовище для швидкої розробки, тестування моделей і спільної роботи.

Python є однією з найпопулярніших мов програмування в галузі науки про дані, машинного навчання та комп'ютерного зору. Вибір Python для цього проєкту був обумовлений кількома ключовими факторами:

1. Широка екосистема бібліотек. Python має розвинену екосистему спеціалізованих бібліотек для роботи з даними та машинного навчання. Для задач обробки зображень і масивів даних було використано декілька основних бібліотек, які суттєво прискорили процес розробки та дозволили зосередитися на суті досліджень.

- NumPy —бібліотека обчислень для Python з відкритим сирцевим кодом. Містить такі структури даних, як багатовимірні масиви та матриці. Може слугувати для виконання певних математичних операцій над масивами, на зразок тригонометричних, статистичних, алгебраїчних обчислень [22];

- Pillow (PIL) — базується на бібліотеці зображень Python Imaging Library (PIL). Ця бібліотека надає широку підтримку форматів файлів, ефективне внутрішнє представлення та досить потужні можливості обробки зображень. Вона охоплює декілька дій з обробки зображень, включаючи операції з точками, фільтрацію, маніпулювання тощо [23];

- OpenCV — одна з найшвидших і найпоширеніших бібліотек для обробки зображень і комп'ютерного зору. Вона підтримується на Github, де понад тисяча учасників роблять свій внесок у розвиток бібліотеки. Створена компанією Intel у 1999 році, вона підтримує багато мов, таких як C, C++, Java та найпопулярнішу Python. OpenCV пропонує близько 2500 алгоритмів для побудови моделей розпізнавання обличчя, виявлення об'єктів, сегментації зображень тощо [24].

2. Глибоке навчання з PyTorch. PyTorch — відкрита бібліотека машинного навчання на основі бібліотеки Torch, що застосовується для задач комп'ютерного бачення та обробки природної мови [25]. Для побудови моделей машинного навчання та глибоких нейронних мереж використовувався фреймворк PyTorch. PyTorch забезпечує динамічну обчислювальну графіку, що робить його надзвичайно зручним для розробки, відладки та тестування моделей. У межах проєкту було використано наступні модулі:

- torch.nn — для побудови архітектури нейронних мереж;
- torch.optim — для вибору та налаштування оптимізаторів, таких як Adam, які використовувалися для навчання моделі;
- torch.utils.data — для роботи з наборами даних, зокрема для їх завантаження та підготовки в батчі для ефективного навчання моделі. Крім того, бібліотека segmentation_models_pytorch полегшила створення моделей для задач сегментації зображень, надаючи готові архітектури, такі як U-Net або FPN, які є загальновизнаними стандартами у сфері комп'ютерного зору.

3. Бібліотеки для аугментації даних. Аугментація зображень — це важливий етап, який дозволяє покращити стійкість моделі до різних варіацій у вхідних даних. У цьому проєкті використовувалася бібліотека Albumentations. Альбументатації це бібліотека Python з відкритим вихідним кодом, розроблена для надання швидких і гнучких можливостей доповнення зображень для практиків машинного навчання. Розроблена командою Albumentations, бібліотека оптимізована для продуктивності та пропонує широкий спектр методів збільшення, включаючи геометричні перетворення, маніпуляції з кольором і розширені розширення, такі як Змішувати та CutMix. Альбументатація сумісна з різними фреймворками глибокого навчання, такими як TensorFlow, PyTorch та Кєрас, що робить його універсальним вибором для проєктів комп'ютерного зору [25].

4. Метрики та візуалізація. Для оцінки роботи моделі та візуалізації результатів застосовувалися:

- `torchmetrics`, зокрема метрика `JaccardIndex`, яка широко використовується в задачах сегментації зображень для оцінки схожості між передбаченими та реальними масками;
- `matplotlib` — бібліотека на мові програмування Python для візуалізації даних двовимірною 2D графікою (3D графіка також підтримується) [25].

Платформа Kaggle була обрана як основне середовище розробки завдяки своїм численним перевагам, які роблять її надзвичайно зручним інструментом для роботи з задачами машинного навчання, аналізу даних та обробки зображень.

Однією з ключових переваг Kaggle є Kaggle Notebooks — інтерактивні середовища для написання та виконання коду. Це дозволяє працювати без необхідності налаштування локального середовища, що суттєво скорочує час на підготовку та технічну організацію роботи. Усі необхідні бібліотеки вже доступні в середовищі, що забезпечує максимальну зручність для швидкого старту роботи. Крім того, Kaggle Notebooks підтримують інтеграцію з потужними інструментами для машинного навчання та глибокого навчання, такими як PyTorch, TensorFlow, Keras та інші. Можливість виконання коду прямо у веб-браузері усуває необхідність завантаження та встановлення додаткового програмного забезпечення на локальний комп'ютер, що особливо корисно для задач із високими вимогами до ресурсів.

Доступ до обчислювальних ресурсів, таких як графічні процесори (GPU) та тензорні обчислювальні блоки (TPU), є однією з ключових переваг сучасних платформ для машинного навчання, таких як Kaggle, Google Colab та інші. Ці ресурси дозволяють суттєво прискорити обчислення завдяки їхній здатності виконувати велику кількість паралельних операцій. Глибокі нейронні мережі, які використовуються для задач комп'ютерного зору, потребують великих обчислювальних ресурсів, і GPU ефективно обробляють матричні операції, що складають основу глибокого навчання, дозволяючи значно скоротити час тренування навіть на великих наборах даних. Робота з великими датасетами, наприклад, зображеннями високої роздільної здатності, вимагає значної

оперативної пам'яті та обчислювальної потужності. GPU забезпечують ефективну обробку таких даних, а TPU спеціалізовані на ще більшому прискоренні тренування, особливо для моделей із великим обсягом параметрів. У рамках проєкту використання GPU стало вирішальним, адже навчання моделей на локальному комп'ютері могло б зайняти значно більше часу або потребувало б потужнішого обладнання. Це дозволило суттєво скоротити час тренування моделей, підвищити продуктивність експериментів і пришвидшити тестування різних гіпотез.

Ще однією важливою особливістю Kaggle є доступ до наборів даних. Платформа пропонує широкий вибір відкритих наборів даних, які можуть бути використані для навчання моделей. Це суттєво зменшує час на пошук відповідних даних та їх попередню обробку, адже більшість наборів вже готові до використання або мають відповідний формат для навчання моделей. У проєкті використовувалися готові набори даних із Kaggle, що дозволило швидше зосередитися на розробці моделей та їх оптимізації. Окрім цього, платформа підтримує можливість завантаження власних наборів даних, що робить її гнучкою для різних дослідницьких потреб [25].

Спільнота Kaggle також є вагомим чинником у виборі цієї платформи. Kaggle надає можливість спілкування з іншими розробниками, участі в обговореннях на форумах, обміну знаннями та отримання порад від більш досвідчених фахівців. Це дає змогу швидко знаходити рішення для проблем, з якими може стикнутися розробник, і сприяє поширенню нових знань та технік. Крім того, платформа дозволяє переглядати чужі ноутбуки та проєкти, що забезпечує прозорість у вирішенні задач та дозволяє використовувати напрацювання інших учасників для власного розвитку. Це також сприяє колаборації між користувачами та репродукції наукових результатів.

Kaggle Learn — це ще одна унікальна особливість платформи, яка пропонує безкоштовні курси та навчальні матеріали з різних тем, пов'язаних із машинним навчанням, обробкою даних та комп'ютерним зором. Завдяки інтерактивним урокам можна швидко отримати необхідні знання та навички для роботи з конкретними технологіями чи методами. У контексті дипломного

проєкту це допомогло ефективно навчитися новим підходам та ознайомитися з найсучаснішими інструментами для обробки зображень та побудови моделей глибокого навчання.

Окрім цього, Kaggle надає змогу брати участь у змаганнях із машинного навчання, де користувачі можуть застосовувати свої навички на практиці та змагатися з іншими учасниками у вирішенні складних завдань. Це чудова можливість для професійного розвитку та отримання зворотного зв'язку від спільноти. Участь у таких змаганнях дозволяє покращити власні навички та побачити, як інші розробники підходять до вирішення аналогічних проблем.

Крім того, завдяки можливості спільного доступу до ноутбуків та проєктів, Kaggle дозволяє ділитися результатами своєї роботи з іншими учасниками, що сприяє репродукції досліджень і перевірці результатів. Це забезпечує прозорість та відкритість у наукових дослідженнях, що робить платформу ідеальною для спільної роботи над проєктами [25].

Завдяки всім цим можливостям, Kaggle є не лише потужним інструментом для розробки та навчання моделей, але й важливою платформою для професійного розвитку та взаємодії з іншими спеціалістами у галузі машинного навчання.

2.3 Вибір оптимальних налаштувань алгоритму для сегментації дахів на аерофотознімках

Для успішного вирішення задачі сегментації дахів було проведено ґрунтовний аналіз різних методологічних підходів, зокрема останніх досягнень у галузі комп'ютерного зору, що зосереджуються на використанні згорткових нейронних мереж (CNN). Ці підходи показують високий рівень ефективності у процесі розпізнавання та сегментації дахів на аерофотознімках. Згорткові нейронні мережі є однією з найсучасніших і найперспективніших технологій для автоматичного аналізу зображень, оскільки вони здатні виявляти складні патерни і деталі об'єктів, зокрема дахів будівель. Однак, як показує дослідження,

навіть ці передові підходи мають певні обмеження, що потрібно враховувати під час розробки та впровадження моделей для сегментації на практиці.

Проведений огляд літератури та аналіз існуючих рішень підкреслюють важливість згорткових нейронних мереж як потужного інструменту для задач сегментації дахів на зображеннях високої роздільної здатності. Основна перевага CNN полягає в здатності автоматично вивчати та ідентифікувати характерні ознаки об'єктів, таких як контури, форми, текстури, що особливо важливо при обробці аерофотознімків, які містять різноманітні будівлі з різними архітектурними особливостями. Ці мережі можуть виявляти патерни, які людина не завжди здатна розпізнати, що робить їх надзвичайно корисними для складних задач, таких як сегментація дахів на великих міських територіях. Однак, незважаючи на значні успіхи CNN, існують певні труднощі, що виникають при застосуванні цих моделей у масштабованих проектах [25].

Одним з основних викликів, що постають перед дослідниками та розробниками, є недостатня точність сегментації у випадках, коли моделі CNN використовуються для аналізу аерофотознімків у великому масштабі. Навіть якщо модель демонструє високі показники точності на зображеннях меншого розміру або на фрагментах зображень, її продуктивність може суттєво знижуватися, коли на вхід подаються масштабні аерофотознімки, що охоплюють великі території. У таких випадках моделі можуть втрачати здатність точно ідентифікувати окремі елементи на зображеннях, що призводить до нечіткого окреслення контурів дахів або взагалі до пропуску деяких об'єктів. Це пов'язано з тим, що моделі, налаштовані для аналізу окремих фрагментів зображення, можуть не враховувати глобальний контекст сцени, що є важливим для правильного виявлення об'єктів на великих масштабах.

Ще однією важливою проблемою при використанні згорткових нейронних мереж (CNN) є їх чутливість до зміни масштабу зображень. Це означає, що коли зображення змінюється, наприклад, збільшується або зменшується, модель може втрачати здатність до точного розпізнавання об'єктів, оскільки її початкові параметри та архітектура були оптимізовані для певного розміру вхідного зображення. Зміна масштабу може спричинити спотворення важливих ознак

зображення, які мережа використовує для класифікації чи сегментації об'єктів, таких як дахи будівель. Це є серйозною проблемою, особливо при обробці аерофотознімків великої території, де об'єкти можуть варіюватися за розміром, формою і контекстом залежно від масштабу зображення.

Для таких задач, як аналіз дахів з аерофотознімків, ця проблема стає ще більш актуальною. Наприклад, зображення на великій території можуть включати як малі будівлі, так і великі структури, і якщо модель не може адаптуватися до різних масштабів, вона може не зуміти точно визначити контури дахів або інші важливі елементи. Це особливо складно, коли зображення охоплює велику площу з різними перспективами та умовами освітлення, що ускладнює виділення правильних ознак для класифікації.

У результаті, при обробці таких великих та варіативних зображень, модель може показувати погані результати у частинах зображення, де масштаби об'єктів сильно відрізняються. Це створює додаткові складнощі, оскільки для досягнення стабільно високої точності на всіх етапах сегментації потрібно враховувати всі можливі варіації в розмірах та пропорціях об'єктів [25].

Щоб подолати цю проблему, однією з можливих стратегій є використання методів, які забезпечують масштабно-інваріантне навчання, таких як багатоаспектні моделі або мережі, здатні адаптуватися до різних масштабів, наприклад, через використання спеціальних шарів для зміни масштабу або використання трансформерів, які можуть враховувати контекст на різних рівнях деталізації. Але навіть із такими підходами проблема зміни масштабу залишатиметься важливою для покращення точності моделей на реальних даних, особливо в складних міських чи природних ландшафтах.

Крім того, CNN можуть демонструвати обмежену продуктивність у складних сценах з високою щільністю об'єктів. Аерофотознімки міських територій часто містять безліч будівель різних форм і розмірів, що ускладнює задачу сегментації. Якщо модель навчена лише на певних типах дахів або на зображеннях з меншою кількістю об'єктів, це може призводити до того, що вона не справлятиметься з виявленням дахів на більш насичених сферах. Наприклад, в умовах, коли дахи будівель розташовані дуже близько один до одного або

накладаються, модель може мати труднощі з розрізненням їх контурів, що погіршує якість сегментації.

Отже, хоча згорткові нейронні мережі демонструють значні переваги у задачах сегментації дахів, їх ефективність значною мірою залежить від якості навчальних даних, налаштувань моделі та особливостей зображень, які використовуються для аналізу. Для подолання зазначених обмежень необхідно доповнювати методи сегментації додатковими підходами післяобробки, що допоможуть поліпшити точність окреслення контурів дахів. Недоліки, у першу чергу, впливають зі складної природи конструкцій даху та мінливості умов зображення, особливо за умов, якщо дахи займають незначну частину зображень, зображено на рисунку 2.3 [26].



Рисунок 2.3 – Приклади з набору даних, де дахи займають незначну частину зображень

Щоб подолати описані вище недоліки та підвищити ефективність таких рішень, що в подальшому дозволить покращити розроблення планів будівель, пропонується вдосконалений підхід. Пропонується об'єднати два ключові компоненти:

- 1) використання моделі детектування DETR від “Meta” [27] на стадії локалізації груп дахів серед повномасштабних зображень;

2) використання моделі на базі ViTs (“Vision Transformers”) [28] для сегментації дахів будівель, оскільки припускається, що досить часто у забудовах використовуються схожі архітектури та види дахів поряд чи в одному мікрорайоні зображено на рисунку 2.3, в такому разі модель сегментації на базі ViTs зможе знаходити схожі види дахів та легко ідентифікувати їх.



а

б

в

Рисунок 2.4 – Приклад з набору даних, де схожі види дахів розташовані в одній щільній забудові

На рисунку 2.4 представлено три типи аерофотозйомок, які відображають різні аспекти міської забудови та земельного використання, що можуть бути корисними для аналізу і планування міської інфраструктури:

а) Аерофотозйомка міської забудови з густим розташуванням житлових будинків: Цей тип знімка демонструє насичену міську середу з високою щільністю житлових будівель, де можна чітко побачити компактне розташування будівель, вуличну мережу та інші елементи міського середовища. Така зйомка дає змогу проводити детальний аналіз структури забудови, виявляючи можливі проблеми з інфраструктурою, такі як обмеження в доступі до громадських просторів або перевантаження транспортних шляхів. Цей тип знімка особливо корисний для досліджень у сфері оптимізації житлових та транспортних мереж.

б) Територія навчального закладу з відкритими просторами та окремими будівлями: На цьому зображенні показана територія навчального закладу, що включає відкриті простори (наприклад, подвір'я, спортивні майданчики, парки)

та окремі будівлі (класні кімнати, адміністративні будівлі тощо). Така зйомка є корисною для оцінки організації простору на кампусі, планування нових будівель чи інфраструктурних змін, а також для визначення зручності розташування різних функціональних зон (освітні, адміністративні, спортивні). Це дозволяє проектувати більш ефективну та зручну середу для навчального процесу.

в) Житлова забудова із середньою щільністю та перехрестям доріг: Цей тип аерофотознімка відображає зону з середньою щільністю житлових будинків та перехрестя доріг, що є важливим елементом для аналізу транспортної мережі. Виявлення таких перехресть може допомогти в оцінці дорожнього руху та можливих проблем з заторами чи безпекою на дорогах. Крім того, це зображення дає змогу побачити, як міські простори взаємодіють між собою, включаючи комунікації між житловими будівлями, торговими та іншими об'єктами інфраструктури, що сприяє покращенню планування нових районів або модернізації існуючих. Таким чином, для комплексного вирішення задачі пошуку та ідентифікації дахів на аерофотознімках пропонується інтеграція моделі DETR у поєднанні з сегментаційними можливостями моделей на основі Vision Transformers (ViTs). Такий підхід дозволить значно підвищити точність і ефективність аналізу зображень, що, у свою чергу, сприятиме покращенню процесу побудови детальних планів будівель.

Останні дослідження та аналіз результатів, зокрема зображень, представлених на рисунках 2.5 та 2.6, демонструють, що моделі класу "Transformer", включаючи DETR та ViTs, є одними з найточніших у завданнях розпізнавання об'єктів і сегментації. Це підтверджується їх здатністю одночасно враховувати глобальний контекст зображення та детальні локальні особливості [29].

Інтеграція DETR дозволяє точно локалізувати об'єкти, що особливо важливо при роботі з великомасштабними зображеннями, такими як аерофотознімки урбаністичних територій. У свою чергу, ViTs забезпечує більш детальну і точну сегментацію дахів, дозволяючи моделі точно визначати контури та структуру будівель. Такий підхід дає можливість створювати більш точні та

комплексні плани забудови, покращуючи якість даних для подальшого використання в архітектурному та інфраструктурному плануванні.

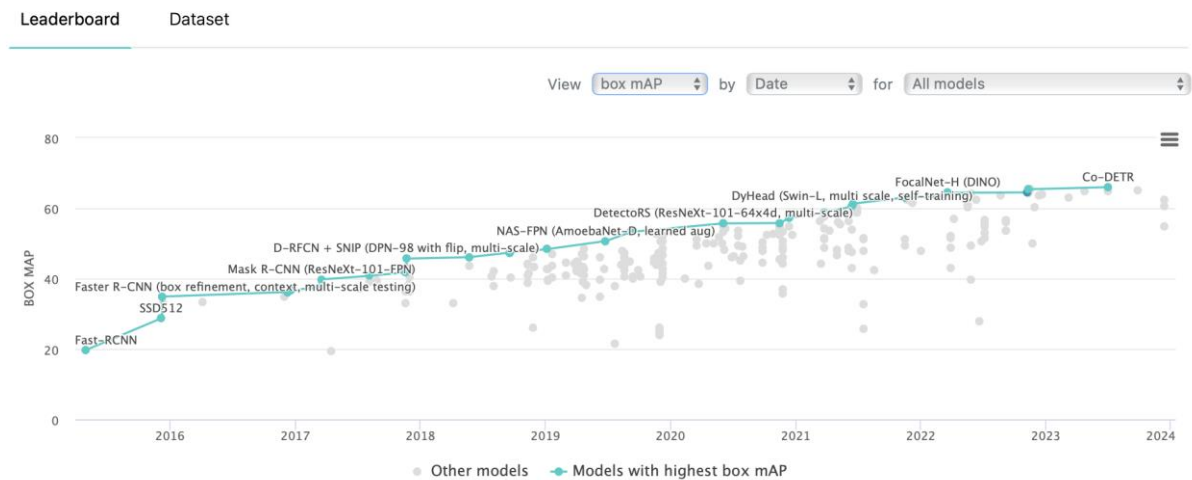


Рисунок 2.5 – Порівняння точності моделей для сегментації зображень [29]

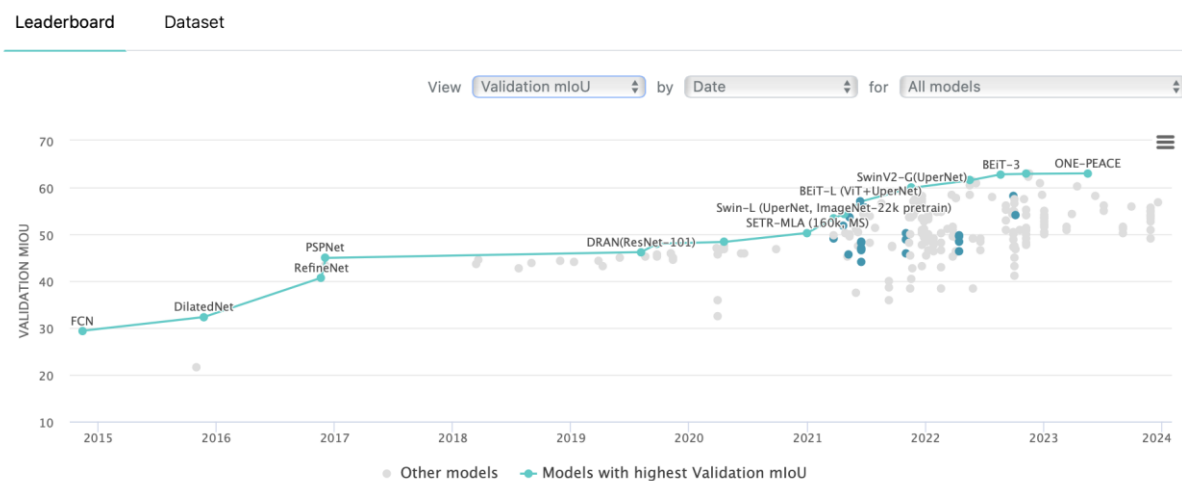


Рисунок 2.6 – Порівняння точності моделей для детектування об’єктів на зображеннях [30] (синім виділено моделі класу “Transformers”)

Запропонований комбінований підхід використовує сильні сторони двох моделей для більш ефективної сегментації дахів на аерофотознімках. Спершу застосовується модель DETR (Detection Transformer), яка спеціалізується на локалізації об’єктів на великих зображеннях. DETR дозволяє виділити групи дахів будівель на масштабних аерофотознімках, що є критичним для роботи з великими територіями та складними урбаністичними сценами. Це забезпечує точне визначення областей, де розташовані будівлі, та дозволяє зосередити

подальший аналіз на цих ділянках. Структурна схема комбінованого підходу з використанням моделей DETR і ViT для сегментації дахів наведена на рисунку 2.7.

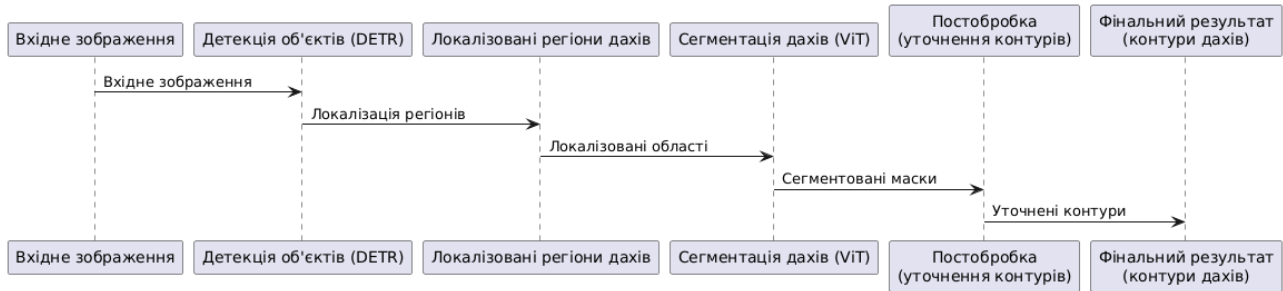


Рисунок 2.7 – Комбінований підхід (DETR + ViT)

Після того, як модель DETR локалізує групи дахів, використовується Vision Transformer (ViT) для детальної сегментації дахів на цих локалізованих ділянках [31]. ViT має високу здатність до обробки складних структур зображень і виявлення тонких відмінностей між об'єктами, тому його застосування є надзвичайно корисним для точного сегментування дахів, навіть якщо вони мають схожі форми та матеріали. Архітектура моделі ViT для сегментації дахів зображена на рисунку 2.8.

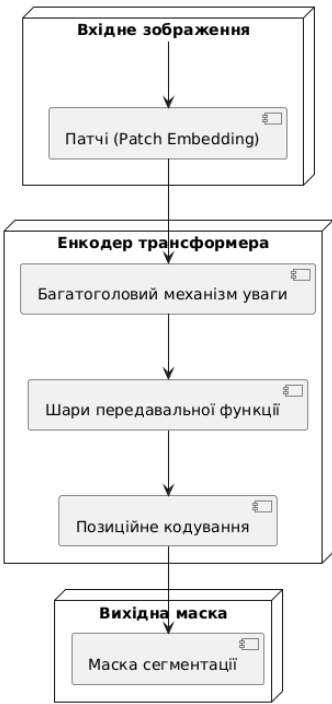


Рисунок 2.8 – Архітектура нейронної мережі (ViT)

ViTs здатні ефективно вивчати глобальний контекст зображень, що допомагає уникнути втрат важливих деталей під час аналізу. Такий підхід дозволяє значно підвищити точність сегментації, оскільки поєднує можливості локалізації великих об'єктів з високоточним аналізом їхніх деталей.

Крім того, запропонований комбінований підхід, що поєднує DETR (Detection Transformer) та ViT (Vision Transformer), має великий потенціал для значного поліпшення процесу створення детальних планів будівель. Точна сегментація дахів є не лише важливим етапом у побудові тривимірних моделей будівель, але й критичним компонентом у створенні архітектурних проектів, розробці містобудівних планів та моніторингу стану міської інфраструктури. Завдяки здатності DETR ефективно працювати з великою кількістю об'єктів на зображеннях та ViT — захоплювати глобальні залежності в зображеннях, комбінований підхід дозволяє досягати більш комплексного та точного аналізу дахів. Це включає не лише загальне визначення їх місцезнаходження, а й глибокий аналіз форми, структури та навіть дрібних деталей кожного даху, що важливо для створення високоточних моделей міської інфраструктури [31].

Такий підхід дозволяє не лише точно розрізняти різноманітні типи дахів і аналізувати їхні геометричні характеристики, але й встановлювати складні взаємозв'язки між дахами та іншими елементами будівельного середовища. Це забезпечує можливість отримувати більш детальну та всебічну інформацію, що є надзвичайно важливим для подальшого використання в архітектурному проектуванні та містобудівництві. Завдяки цьому архітектори й інженери можуть приймати більш обґрунтовані рішення щодо оптимального використання простору, враховуючи як технічні, так і естетичні аспекти.

Крім того, запропонований метод забезпечує ефективніший моніторинг змін у міській інфраструктурі. Це стає особливо важливим у контексті швидкої урбанізації, коли міста активно розвиваються, а вимоги до їхньої стійкості та екологічної ефективності постійно зростають. Такий підхід дозволяє виявляти динаміку змін у структурі міських територій, прогнозувати їхній розвиток та

вчасно реагувати на потенційні проблеми, що виникають у процесі розширення міських зон.

Комбіноване використання DETR (DEtection TRansformer) і ViT (Vision Transformer) значно підвищує точність визначення об'єктів на складних аерофотознімках, що відрізняються великою кількістю деталей і неоднорідністю структур. Завдяки цьому зменшується ймовірність помилок, що особливо важливо для застосування таких технологій у системах автоматизованого планування, управління міським розвитком та адаптивних системах управління міською інфраструктурою. Складність таких зображень вимагає алгоритмів, здатних не лише ідентифікувати об'єкти, але й враховувати контекст та міжоб'єктні взаємозв'язки, що DETR і ViT виконують надзвичайно ефективно [31].

Завдяки своїм перевагам, таким як висока точність, зниження похибок та можливість роботи з великими наборами даних, цей підхід має великий потенціал для широкого застосування. Його можна інтегрувати в урбаністичні дослідження для вивчення міських процесів, розробки стратегії розвитку територій, моніторингу стану будівель та інфраструктури. У сфері автоматизованого планування цей підхід може слугувати основою для створення інтерактивних карт, побудови 3D-моделей міських територій та оптимізації процесів проектування. У результаті це сприятиме покращенню ефективності містобудівного процесу, забезпечуючи більш точний і надійний інструмент для підтримки прийняття рішень.

2.4 Висновки

У рамках попереднього аналізу даних була проведена комплексна оцінка вхідної інформації, яка включає аерофотознімки разом з масками сегментації дахів. Було детально проаналізовано ключові види дахів, що дало змогу підготувати якісний датасет для подальших досліджень та розробки моделей.

У процесі підготовки даних були ретельно розглянуті етапи передобробки, що забезпечують належну підготовку зображень для ефективного навчання

моделей машинного навчання. Це включає в себе корекцію, нормалізацію та збільшення даних, щоб підвищити загальну якість вхідної інформації.

Для вибору оптимальних технологій обробки зображень були проаналізовані різноманітні методології, включаючи згорткові нейронні мережі (CNN) та трансформери. В результаті обрано модель DETR (DEtection TRansformer) для локалізації груп дахів, яка забезпечує стійкість та точність при виявленні об'єктів, а також модель ViTs (Vision Transformers) для сегментації дахів, котра демонструє високу ефективність у завданнях з розпізнавання зображень.

Крім того, в процесі реалізації проекту було розглянуто методи чистки та післяобробки зображень, такі як алгоритм пошуку контурів Canny Edge Detection. Цей метод дозволяє значно покращити точність ідентифікації контурів дахів, що є критично важливим для подальшої обробки та аналізу отриманих результатів.

Таким чином, даний розділ не лише закладає надійний фундамент для побудови моделей машинного навчання, але й підкреслює важливість ретельного підходу до попереднього аналізу даних і вибору найбільш чітких та ефективних технологій обробки зображень. Це дозволить досягти максимальної точності в ідентифікації дахів на аерофотознімках, відкриваючи нові можливості для подальшого розвитку цієї області досліджень.

3 РОЗРОБЛЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ТЕХНОЛОГІЇ ПОБУДОВИ ПЛАНУ БУДІВЕЛЬ ПО АЕРОФОТОЗЙОМЦІ ЇХ ДАХІВ

3.1 Алгоритм вирішення задачі

Отже, для розв'язання поставленої задачі пропонується удосконалена інтелектуальна технологія, архітектура якої складається із трьох невід'ємних компонентів, призначених для підвищення точності та ефективності всього процесу.

По-перше, запропонована технологія включає в себе механізм локалізації, призначений для ідентифікації груп дахів на великомасштабних зображеннях. Даний початковий крок має ключове значення для виділення регіонів інтересу шляхом зосередження виключно на відповідних областях. Використовуючи моделі детекції об'єктів, такі як DETR, система дозволяє визначити скупчення дахів серед складних міських ландшафтів або різних умов зображення.

По-друге, після завершення фази локалізації запропонована архітектура переходить до наступного ключового етапу — сегментації, що відіграє вирішальну роль у створенні точних і детальних масок дахів. Для цього використовується сучасна модель сегментації, яка поєднує глибоке навчання та передові методи обробки зображень. Завдяки цьому етапу досягається високий рівень деталізації, необхідний для виділення кожного даху навіть у складних міських умовах.

Застосування трансформерів (Vision Transformers, ViTs), які спеціально навчені для сегментації дахів, дає змогу моделі ефективно обробляти особливості різних дахів, враховуючи їхню форму, текстуру та можливі варіації освітлення. ViTs дозволяють системі захоплювати складні геометричні деталі, а також точно визначати межі дахів, навіть якщо вони частково перекриті іншими об'єктами або злиті з фоном. Завдяки цим можливостям модель сегментації не тільки формує точні контури дахів, але й здатна розрізняти окремі дахи в тісній забудові чи на багатошарових знімках.

Цей етап є критичним для подальшого аналізу, оскільки створені маски сегментації забезпечують основу для таких завдань, як оцінка площі дахів, ідентифікація їхніх матеріалів чи планування реконструкції.

Процес сегментації дахів, а також інтеграція цього алгоритму у загальну архітектуру, чітко відображені у блок-схемі алгоритму сегментації дахів, що наведена на рисунку 3.1. Дана схема ілюструє взаємозв'язок між попередньою обробкою даних, фазою локалізації та сегментацією, дозволяючи зрозуміти всі етапи трансформації вхідних зображень у готові для використання результати..

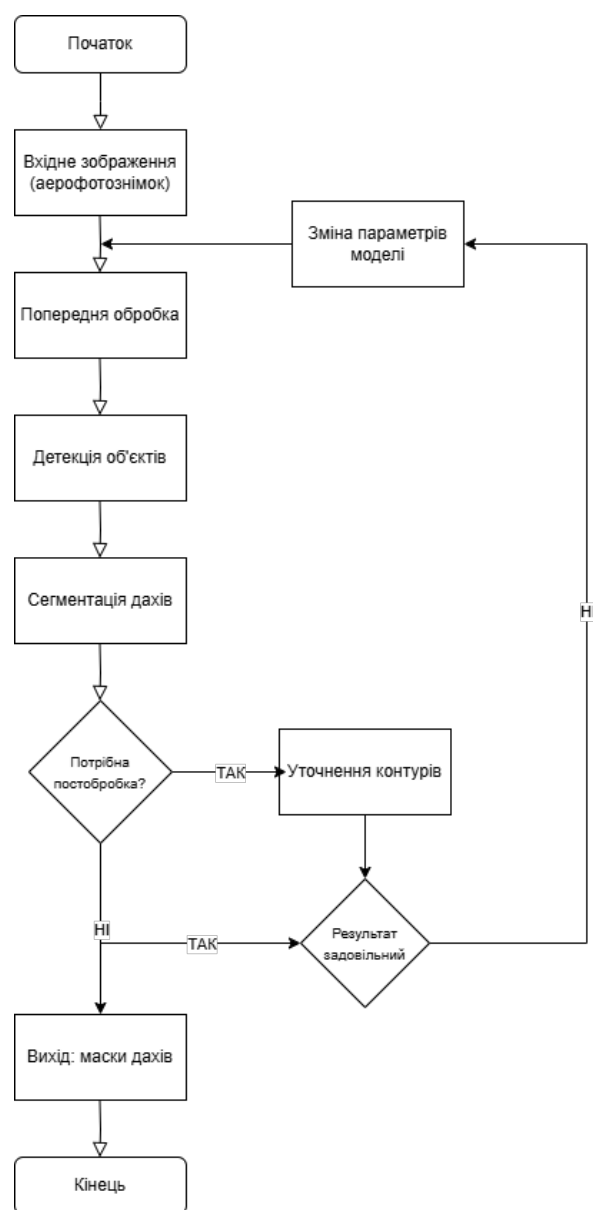


Рисунок 3.1 – Блок-схема алгоритму сегментації дахів

Приклад структурної схеми запропонованої архітектури, що включає в себе описані вище компоненти зображено на рисунку 3.2.

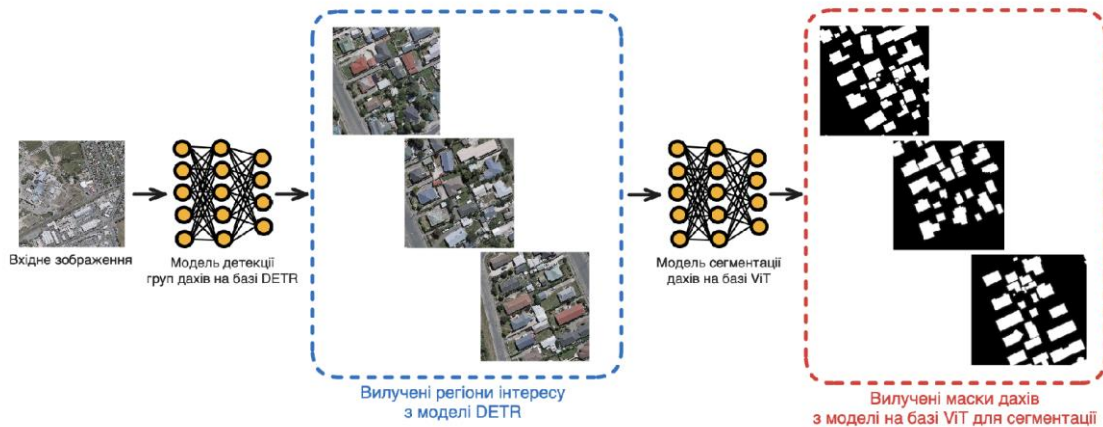


Рисунок 3.2 – Структурна схема архітектури інтелектуальної технології, що включає компоненти з використанням моделей DETR та ViTs

Для оцінювання ефективності роботи моделі детекції об’єктів DETR та сегментаційної моделі на базі ViTs пропонується використати Коефіцієнт Жаккара (Jaccard index або IoU - Intersection over Union) в якості основної метрики, оскільки даний показник кількісно визначає просторове перекриття між прогнозованою та справжньою обмежувальною рамкою/сегментаційною маскою зі значеннями в діапазоні від 0 (без перекриття) до 1 (ідеальне перекриття):

$$\text{IoU}(A,B)=\frac{|A\cap B|}{|A\cup B|}, \quad (3.1)$$

де A - представляє обмежуючу рамку або множину з фактичними даними;
B - представляє передбачену обмежуючу рамку або множину.

Загальну взаємодію між компонентами інтелектуальної технології, що включають механізм локалізації на основі DETR, модель сегментації на базі ViTs, а також методи післяобробки, що сприяють точному визначенню контурів дахів та їх подальшому аналізу.

Зображено на рисунку 3.3.



Рисунок 3.3 – Взаємодія моделей та інструментів

По-третє, на додаток до компонентів локалізації та сегментації, запропонована технологія включає ряд класичних методів післяоброблення для подальшого вдосконалення та покращення результатів аналізу. Ключовим серед цих методів є застосування класичних алгоритмів пошуку контурів Canny Edge Detection, які дозволяють точно ідентифікувати контури даху та лінії гребенів даху для побудови планів будівель. Використовуючи методи контурної фільтрації, система може розрізняти відповідні елементи даху та сторонні

елементи, удосконалюючи таким чином інтелектуальну технологію та підвищуючи її точність.

Зображено на рисунку 3.4.

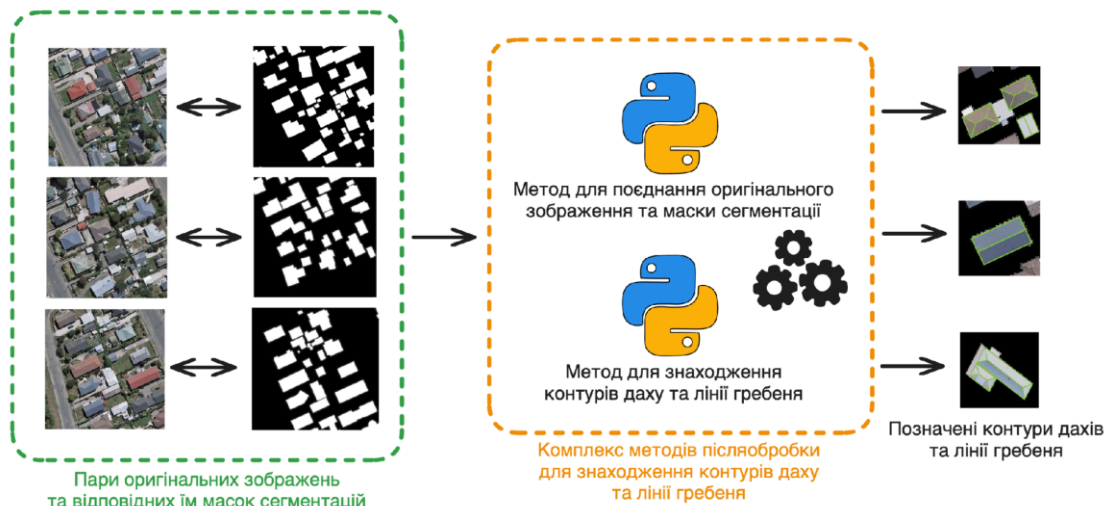


Рисунок 3.4 – Структурна схема архітектури інтелектуальної технології, що включає компоненти з використанням методів післяоброблення для знаходження контурів дахів та ліній гребеня

Запропонована інтелектуальна технологія представляє собою цілісний підхід до вирішення задачі створення плану будівель за даними аерофотозйомки. Вона базується на ефективному поєднанні сучасних методів локалізації, таких як DETR, моделей сегментації на основі ViTs, а також класичних методів післяоброблення, що забезпечує високу адаптивність до різноманітних умов і характеристик аерофотознімків. Використання DETR дозволяє точно визначати межі будівельних об'єктів, а впровадження моделей на основі ViTs забезпечує глибокий аналіз і деталізацію контурів дахів, навіть у випадках складних геометричних форм.

Крім того, класичні методи післяоброблення сприяють уточненню результатів сегментації та дозволяють досягти оптимальної якості аналізу. Інтеграція цих підходів у єдину архітектуру створює комплексне рішення, яке не тільки дозволяє підвищити точність ідентифікації координат фундаменту

будівель, але й забезпечує масштабованість технології до роботи з великими наборами даних аерофотозйомки.

Така технологія спрямована на розширення можливостей автоматизації містобудівного планування, підвищення ефективності аналізу урбаністичних територій і створення сучасних інструментів для моніторингу та управління міськими ресурсами. Результати запропонованого підходу можуть знайти застосування в геоінформаційних системах, системах автоматизованого проєктування та інструментах для створення цифрових моделей міст.

3.2 Підготовка до роботи

На рисунку 3.5 ілюструється імпорт всіх бібліотек.

```
In [2]: import os
import sys

from tqdm import tqdm
import time
from datetime import timedelta

import random
import numpy as np

from PIL import Image
import cv2

import torch
import torch.nn as nn
import segmentation_models_pytorch as smp
from torch.utils.data import Dataset, DataLoader
import torch.optim as optim

import albumentations as A

from torchmetrics import JaccardIndex
import matplotlib.pyplot as plt
```

Рисунок 3.5 – Імпорт бібліотек

У фрагменті коду зображеному на рисунку 3.6 визначається пристрій для обчислень (CPU або GPU), який буде використовуватися для виконання подальших операцій в нейронній мережі. Змінна `device` встановлюється на значення `"cuda"` у випадку, якщо GPU (CUDA) доступний на системі, інакше використовується центральний процесор (`cpu`). Це забезпечує гнучкість, оскільки модель автоматично адаптується до апаратного середовища, що

доступне для тренування або інференсу, тим самим покращуючи продуктивність на GPU або зберігаючи сумісність із CPU.

```
In [3]: device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

Рисунок 3.6 – Визначення пристрою обчислень

На рисунку 3.7 зображено результат виконання команди:

```
In [4]: print(f"Device: {device}")
```

Device: cuda

Рисунок 3.7 – Процесор на якому буде виконуватись

Виведення показує, що обчислення будуть виконуватись на графічному процесорі (GPU), оскільки в результаті виведено Device: cuda. Це означає, що система підтримує технологію CUDA, і PyTorch успішно виявив доступний графічний процесор для прискорення обчислень.

Використання GPU значно підвищує продуктивність при тренуванні великих моделей машинного навчання, особливо в задачах глибокого навчання, де обчислювальні ресурси CPU можуть виявитися недостатніми. Таким чином, забезпечується ефективніше використання обчислювальних ресурсів системи, що дозволяє зменшити час тренування моделей.

Такий результат підтверджує, що обчислювальний процес буде виконуватися на GPU, що є бажаним результатом для задач із великими обчислювальними навантаженнями.

Налаштування гіперпараметрів нейронної мережі зображено на рисунку 3.8.

```
In [5]: LEARNING_RATE = 0.001  
        BATCH_SIZE = 2  
        ENCODER_NAME = "mit_b2"  
        ENCODER_WEIGHTS = "imagenet"
```

Рисунок 3.8 – Налаштування гіперпараметрів нейронної мережі

`LEARNING_RATE = 0.001` — встановлено швидкість навчання моделі на рівні 0.001. Це значення визначає, як швидко модель буде оновлювати свої ваги під час навчання, впливаючи на процес оптимізації.

`BATCH_SIZE = 2` — розмір батчу становить 2. Це означає, що на кожній ітерації навчання модель отримує два зразки для обчислення градієнтів і оновлення ваг.

`ENCODER_NAME = "mit_b2"` — використовується енкодер під назвою `mit_b2`, який є частиною архітектури моделі, що відповідає за початкове представлення даних перед передачею їх до основних шарів нейронної мережі.

`ENCODER_WEIGHTS = "imagenet"` — ваги енкодера ініціалізовано за допомогою попередньо натренованої моделі на наборі даних ImageNet, що дозволяє моделі краще розпізнавати загальні риси зображень на ранніх етапах навчання.

Ці налаштування є важливими для забезпечення стабільного та ефективного навчання моделі.

Представлено код функції, що виконує обробку зображення даху на основі сегментаційної маски. Функція `roof_processing` виконує кілька етапів:

1. `original_image` та `segmentation_mask` — вхідні параметри, де перше є оригінальним зображенням, а друге — сегментаційною маскою, що виділяє область даху на зображенні;

2. `masked_image = cv.bitwise_and(original_image, segmentation_mask, mask=segmentation_mask)` — застосовується побітове "І" для виділення області даху з оригінального зображення за допомогою сегментаційної маски. Це дозволяє обмежити подальшу обробку лише областю даху;

3. `grayscale_image = cv2.cvtColor(masked_image, cv2.COLOR_BGR2GRAY)` — перетворення кольорового зображення у відтінки сірого, що є необхідним для подальшої обробки зображення, зокрема для виділення контурів;

4. `blurred_image = cv2.GaussianBlur(grayscale_image, kernel_size, 0)` — згладження зображення за допомогою гаусівського розмивання для зменшення

шумів і поліпшення якості виявлення контурів. Параметр `kernel_size=(3, 3)` визначає розмір ядра фільтра;

5. `edged_image = cv2.Canny(blurred_image, lower_limit, upper_limit)` — застосовується алгоритм Canny для виявлення контурів. Параметри `lower_limit=1` і `upper_limit=50` задають порогові значення для визначення меж контурів;

6. `return edged_image` — результатом функції є зображення з виділеними контурами області даху, яке буде передане для подальшого аналізу або візуалізації.

Цей код є частиною обробки зображень для виділення контурів даху на основі попередньо створеної сегментаційної маски зображено на рисунку 3.9.

```
In [6]: def roof_processing(original_image, segmentation_mask, kernel_size=(3, 3), lower_limit=1, upper_limit=50):
masked_image = cv.bitwise_and(original_image, segmentation_mask, mask=segmentation_mask)
grayscale_image = cv2.cvtColor(masked_image, cv2.COLOR_BGR2GRAY)
blurred_image = cv2.GaussianBlur(grayscale_image, kernel_size, 0)
edged_image = cv2.Canny(blurred_image, lower_limit, upper_limit)
return edged_image
```

Рисунок 3.9 – Процес обробки даху на основі сегментаційної маски

Представлений код функції `roof_ridge_detection`, яка виконує виявлення ребер (ліній) даху на зображенні за допомогою алгоритму ліній Хафа. Основні етапи роботи функції:

1. `original_image` та `edged_image` — вхідні параметри. Перше — це оригінальне зображення, а друге — зображення з виділеними контурами, яке попередньо оброблене за допомогою функції виявлення контурів (наприклад, Canny);

2. `hough_grid=1`, `threshold=15`, `min_line_length=50`, `max_line_gap=30` — параметри алгоритму HoughLinesP;

3. `hough_grid` — параметр сітки для Hough-трансформації, що визначає крок перевірки можливих ліній;

4. `threshold` — мінімальна кількість точок на лінії, необхідних для її виявлення;

5. `min_line_length` — мінімальна довжина лінії для її розпізнавання як ребра даху;
6. `max_line_gap` — максимальна відстань між точками на лінії, яку алгоритм може об'єднати в одну лінію;
7. `theta = np.pi / 180` — кутова розгортка для пошуку ліній, що визначає розділення кутів у радіанах;
8. `lines = cv2.HoughLinesP(edged_image, hough_grid, theta, threshold, np.array([]), min_line_length, max_line_gap)` — основна частина алгоритму Хафа для виявлення ліній на зображенні з контуром. Повертає список ліній, кожна з яких визначається координатами її початкової та кінцевої точок;
9. `for line in lines:` — цикл, що проходить по всіх виявлених лініях. Для кожної лінії за допомогою функції `cv2.line()` малюється зелена лінія (колір (0, 255, 0)) на оригінальному зображенні, використовуючи координати кінців ліній (`x1, y1, x2, y2`);
10. `return original_image` — повертається оригінальне зображення з накресленими лініями, що відповідають ребрам даху.

Ця функція є важливою частиною алгоритму аналізу зображень, зокрема для виявлення архітектурних елементів даху, таких як ребра чи кромки, на основі контурів, знайдених раніше зображений на рисунку 3.10.

```
In [7]: def roof_ridge_detection(original_image, edged_image, hough_grid=1, threshold=15, min_line_length=50, max_line_gap=30):
        theta = np.pi / 180
        lines = cv2.HoughLinesP(edged_image, hough_grid, theta, threshold, np.array([]), min_line_length, max_line_gap)

        for line in lines:
            for x1, y1, x2, y2 in line:
                cv2.line(original_image, (x1, y1), (x2, y2), (0, 255, 0), 3)

        return original_image
```

Рисунок 3.10 – Виявлення ребер даху за допомогою ліній Хафа

Представлена функція `seed_everything`, яка дозволяє зафіксувати початковий стан генераторів випадкових чисел у різних бібліотеках. Це необхідно для того, щоб кожен запуск коду призводив до однакових результатів,

що важливо для відтворюваності експериментів у машинному навчанні. Основні кроки:

1. `random.seed(seed)` — задає початковий стан для модуля `random`, який використовується для роботи з випадковими числами в Python;
2. `os.environ["PYTHONHASHSEED"] = str(seed)` — встановлює `seed` для хеш-функцій Python, щоб порядок хешованих елементів залишався сталим між різними запусками програми;
3. `np.random.seed(seed)` — задає випадковий стан для генератора випадкових чисел бібліотеки NumPy, яка широко застосовується для математичних обчислень і роботи з масивами даних;
4. `torch.manual_seed(seed)` і `torch.cuda.manual_seed(seed)` — встановлюють однаковий `seed` для обчислень в бібліотеці PyTorch як для процесора (CPU), так і для графічного процесора (GPU). Це гарантує, що модель навчання буде давати однакові результати незалежно від платформи;
5. `torch.backends.cudnn.deterministic = True` — перемикає режим обчислень cuDNN (модуль для прискорення нейронних мереж) у детерміністичний режим для забезпечення стабільних результатів;
6. `torch.backends.cudnn.benchmark = False` — вимикає режим вибору найоптимальніших параметрів для обчислень, оскільки це може змінювати результати між різними запусками.

Виклик `seed_everything(42)` задає значення `seed` 42, що часто використовується для узгодженості під час тестування та проведення експериментів зображень на рисунку 3.11.

```
In [8]: def seed_everything(seed):
        random.seed(seed)
        os.environ["PYTHONHASHSEED"] = str(seed)
        np.random.seed(seed)
        torch.manual_seed(seed)
        torch.cuda.manual_seed(seed)
        torch.backends.cudnn.deterministic = True
        torch.backends.cudnn.benchmark = False

        seed_everything(42)
```

Рисунок 3.11 – Ініціалізація випадкового стану для забезпечення відтворюваності результатів

Представлено реалізацію класу `DataRetriever`, який успадковується від `Dataset` (базового класу PyTorch для роботи з наборами даних). Цей клас призначений для завантаження зображень і відповідних масок сегментації з директорії та їхньої обробки перед подачею в модель. Він є частиною підготовки даних для завдань сегментації зображень, таких як розпізнавання об'єктів. Основні компоненти класу:

1. `init(self, data_dir, data_list, transform=None)` — ініціалізує клас;
2. `data_dir` — шлях до директорії з даними;
3. `data_list` — список імен файлів із зображеннями;
4. `transform` — функція для виконання аугментації та перетворення зображень і масок (якщо застосовується);
5. `len(self)` — повертає кількість зразків у наборі даних, що дорівнює довжині списку `data_list`;
6. `getitem(self, idx)` — повертає зображення та відповідну маску сегментації для заданого індексу `idx`.

Якщо індекс поданий у форматі `torch.Tensor`, його перетворюють на список.

В залежності від того, чи дані належать до навчального (`train`) або валідаційного (`val`) набору, зображення та маска завантажуються за різними шляхами.

`image_path` та `mask_path` визначають шляхи до відповідних файлів зображення та маски.

Маска сегментації завантажувється як одноканальне зображення (градація відтінків сірого) і нормалізується шляхом ділення на 255.

Якщо задана аугментація `transform`, то вона застосовується до зображення та маски. Перетворення перетворюють зображення у формат `torch.Tensor`, переставляючи його канали (щоб відповідати формату PyTorch) та додають маску до тензорів, що відповідають формату сегментації.

Метод повертає словник з двома ключами. `"X"` — тензор зображення. `"y"` — тензор маски сегментації.

Цей клас дозволяє зручно організувати завантаження зображень та їх відповідних масок, а також готувати їх до використання в нейронній мережі для завдань комп'ютерного зору зображено на рисунку 3.12.

```
In [9]:
class DataRetriever(Dataset):
    def __init__(self, data_dir, data_list, transform=None):
        self.data_dir = data_dir
        self.data_list = data_list
        self.transform = transform

    def __len__(self):
        return len(self.data_list)

    def __getitem__(self, idx):
        if torch.is_tensor(idx):
            idx = idx.tolist()

        # image_path = f"{self.data_dir}/image/{self.data_list[idx]}"
        # image = np.array(Image.open(image_path).convert("RGB"))

        # print(self.data_dir)

        if "train" in self.data_dir:
            image_path = f"{self.data_dir}/image/{self.data_list[idx]}"
            mask_path = f"{self.data_dir}/label/{self.data_list[idx].split('.')[0]}_vis.tif"
        elif "val" in self.data_dir:
            image_path = f"{self.data_dir}/image/{self.data_list[idx]}.tif"
            mask_path = f"{self.data_dir}/label/{self.data_list[idx]}_vis.tif"

        image = np.array(Image.open(image_path).convert("RGB"))
        mask = np.array(Image.open(mask_path).convert("L"))
        mask = mask / 255.0

        if self.transform:
            augmentations = self.transform(image=image, mask=mask)
            image = torch.from_numpy(augmentations["image"]).permute(2, 0, 1)
            mask = torch.from_numpy(augmentations["mask"]).int().unsqueeze(0)

        return {"X": image, "y": mask}
```

Рисунок 3.12 – Клас для завантаження та обробки даних із масками сегментації

Налаштування шляхів до основних директорій та текстових файлів, що містять списки зображень для навчання, валідації та тестування в задачі сегментації дахів на аерофотознімках.

1. DATA_DIR = "/kaggle/input/aerialimageryforroofsegmentation/" — основний шлях до директорії з даними, що містить зображення для задачі сегментації дахів. Ця директорія є кореневою для інших піддиректорій та текстових файлів;

2. TRAIN_DATA_DIR = "/kaggle/input/aerialimageryforroofsegmentation/train" — шлях до набору навчальних даних. Він вказує на директорію, де зберігаються зображення та

маски, що використовуються для тренування моделі;

3. `VALID_DATA_DIR` — шлях до набору валідаційних даних. Валідаційний набір використовують для перевірки точності моделі під час тренування, щоб оцінити її продуктивність на нових даних;

4. `TRAIN_TXT` = `f"{DATA_DIR}/train.txt"` — шлях до текстового файлу, що містить список імен зображень, використаних у навчальному наборі. Цей файл вказує, які зображення та маски використовуються на етапі тренування;

5. `VALID_TXT` = `f"{DATA_DIR}/val.txt"` — шлях до текстового файлу, що містить список імен зображень для валідації. Він допомагає визначити, які зображення і маски повинні бути завантажені для оцінки моделі під час тренувального процесу;

6. `TEST_TXT` = `"/kaggle/input/aerial-imagery-data-for-training/test.txt"` — шлях до тестового набору даних, який зберігається окремо. Цей файл містить імена зображень, які будуть використані для остаточної перевірки продуктивності моделі після тренування і валідації.

Ці змінні визначають шляхи до всіх необхідних даних, що використовуються для тренування та тестування моделі сегментації дахів на основі аерофотознімків, і забезпечують організовану структуру файлів зображено на рисунку 3.13.

```
In [10]:
DATA_DIR = "/kaggle/input/aerialimageryforroofsegmentation/"
TRAIN_DATA_DIR = "/kaggle/input/aerialimageryforroofsegmentation/train"
VALID_DATA_DIR = "/kaggle/input/aerialimageryforroofsegmentation/val"

TRAIN_TXT = f"{DATA_DIR}/train.txt"
VALID_TXT = f"{DATA_DIR}/val.txt"
TEST_TXT = "/kaggle/input/aerial-imagery-data-for-training/test.txt"
```

Рисунок 3.13 – Визначення шляхів до наборів даних для сегментації дахів

Код, який завантажує списки імен файлів із текстових файлів для наборів даних тренування та валідації. Ці списки використовуються для визначення, які

зображення і відповідні маски сегментації повинні бути завантажені на кожному етапі обробки даних:

1. `TRAIN_LIST = []` та `VALID_LIST = []` — ініціалізація порожніх списків для зберігання імен файлів, що використовуються відповідно для тренувальних та валідаційних наборів даних;
2. `with open(TRAIN_TXT, 'r') as file:` — відкриває файл `train.txt`, що містить список імен зображень для навчання, у режимі читання;
3. `TRAIN_LIST = [line.strip() for line in file.readlines()]` — читає всі рядки з файлу та видаляє зайві пробіли (якщо є) за допомогою `strip()`. Кожен рядок представляє ім'я зображення, яке додається до списку `TRAIN_LIST`;
4. `with open(VALID_TXT, 'r') as file:` — аналогічно відкриває файл `val.txt`, що містить список зображень для валідації;
5. `VALID_LIST = [line.strip() for line in file.readlines()]` — зчитує і зберігає всі імена зображень у список `VALID_LIST` після очищення від пробілів.

Коментарі, які закоментовані (код для тестового набору):

1. `# TEST_LIST = []` — ініціалізація порожнього списку для тестових даних;
2. `# with open(TEST_TXT, 'r') as file:` — за потреби можна розкоментувати цей код для читання списку зображень для тестового набору з файлу `test.txt`;
3. `# TEST_LIST = [line.strip() for line in file.readlines()]` — аналогічно зчитуються імена файлів для тестових даних.

Цей код забезпечує зручне завантаження імен зображень для подальшої роботи з тренувальним і валідаційним набором даних у задачі сегментації зображень. Він автоматизує процес підготовки даних, що є важливим кроком для ефективного навчання моделі. Завдяки такому підходу, всі зображення можна швидко завантажити та організувати в необхідний формат для подальшої обробки, що значно покращує ефективність і швидкість розробки моделі. Цей етап також сприяє точнішій валідації результатів, оскільки забезпечує коректне розподілення даних на тренувальну і тестову вибірки. Підготовка даних є

важливою частиною процесу, оскільки вона дозволяє досягти високих результатів у задачах комп'ютерного зору зображено на рисунку 3.14.

```
In [11]:
TRAIN_LIST = []
VALID_LIST = []
# TEST_LIST = []

# Open the file in read mode
with open(TRAIN_TXT, 'r') as file:
    # Read all lines from the file and store them in the list
    TRAIN_LIST = [line.strip() for line in file.readlines()]

# Open the file in read mode
with open(VALID_TXT, 'r') as file:
    # Read all lines from the file and store them in the list
    VALID_LIST = [line.strip() for line in file.readlines()]

# # Open the file in read mode
# # with open(TEST_TXT, 'r') as file:
# #     # Read all lines from the file and store them in the list
# #     TEST_LIST = [line.strip() for line in file.readlines()]
```

Рисунок 3.14 – Завантаження списків зображень для тренування та валідації

На рисунку 3.15 представлено код, який задає аугментації для навчального та валідаційного наборів даних у процесі сегментації зображень. Аугментації використовуються для поліпшення якості тренування моделі та підвищення її здатності до узагальнення на нових даних.

```
In [12]:
# Define transforms for data augmentation
train_augs = A.Compose([
    #
    A.Resize(512, 512),
    A.RandomCrop(always_apply=True, p=1.0, height=512, width=512),
    #
    A.GaussianBlur(),
    #
    A.RandomBrightnessContrast(),
    #
    A.HueSaturationValue(),
    #
    A.HorizontalFlip(),
    A.Normalize(
        mean=[0.0, 0.0, 0.0],
        std=[1.0, 1.0, 1.0],
        max_pixel_value=255.0,
        always_apply=True
    ),
],
)

valid_augs = A.Compose([
    #
    A.Resize(512, 512),
    A.RandomCrop(always_apply=True, p=1.0, height=512, width=512),
    A.Normalize(
        mean=[0.0, 0.0, 0.0],
        std=[1.0, 1.0, 1.0],
        max_pixel_value=255.0,
        always_apply=True
    ),
],
)
```

Рисунок 3.15 – Визначення аугментацій для навчального та валідаційного наборів даних

Бібліотека `Albumentations` (позначена як `A`) застосовується для виконання аугментацій:

1. `train_augs` — набір аугментацій, застосовуваних до тренувальних зображень;
2. `A.RandomCrop(always_apply=True, p=1.0, height=512, width=512)` — випадкове обрізання (`crop`) зображень до розміру 512x512 пікселів. Цей метод застосовується до кожного зображення незалежно від його вихідного розміру, що дозволяє забезпечити сталість розміру зображень на вході в модель;
3. `A.Normalize(mean=[0.0, 0.0, 0.0], std=[1.0, 1.0, 1.0], max_pixel_value=255.0, always_apply=True)` — нормалізація зображення, яка перетворює пікселі на діапазон `[0, 1]` на основі значень середнього та стандартного відхилення. Тут використовується стандартна нормалізація, яка не змінює самих значень (середнє значення 0.0, стандартне відхилення 1.0). Це важливий етап підготовки зображень для роботи з моделями глибокого навчання;
4. `valid_augs` — набір аугментацій, застосовуваних до валідаційних зображень;
5. `A.RandomCrop(always_apply=True, p=1.0, height=512, width=512)` — так само, як і для навчальних даних, виконується випадкове обрізання зображення до розміру 512x512 пікселів;
6. `A.Normalize(mean=[0.0, 0.0, 0.0], std=[1.0, 1.0, 1.0], max_pixel_value=255.0, always_apply=True)` — нормалізація аналогічна до тренувального набору. Це дозволяє забезпечити однакові умови під час обробки зображень у валідаційному наборі.

Деякі аугментації, такі як `GaussianBlur`, `RandomBrightnessContrast`, `HueSaturationValue`, та `HorizontalFlip`, закоментовані, але їх можна використовувати для додаткової варіативності даних під час навчання.

У цьому фрагменті коду відбувається ініціалізація двох основних наборів даних для машинного навчання: `train_dataset` (навчальний набір даних) та `valid_dataset` (валідаційний набір даних).

`train_dataset` створюється за допомогою класу `DataRetriever`, який приймає три параметри:

1. `data_dir=TRAIN_DATA_DIR`: вказує на директорію з навчальними даними;
2. `data_list=TRAIN_LIST`: список випадків для навчання;
3. `transform=train_augs`: трансформації та аугментації для покращення генералізації моделі.

`valid_dataset` налаштовується аналогічним чином, але для валідаційного набору:

1. `data_dir=VALID_DATA_DIR`: директорія з валідаційними даними;
2. `data_list=VALID_LIST`: список випадків для валідації;
3. `transform=valid_augs`: специфічні трансформації для валідаційних даних.

Це дозволяє забезпечити розподіл даних для навчання та перевірки моделі, що є стандартною практикою для оцінки її продуктивності та узагальнення на нових даних зображено на рисунку 3.16.

```
In [13]: # Create the train dataset
#train_dataset = DataRetriever(data_dir=DATA_DIR, cases_list=TRAIN_LIST, transform=train_augs)
train_dataset = DataRetriever(data_dir=TRAIN_DATA_DIR, data_list=TRAIN_LIST, transform=train_augs)

# Create the valid dataset
#valid_dataset = DataRetriever(data_dir=DATA_DIR, cases_list=VALID_LIST, transform=valid_augs)
valid_dataset = DataRetriever(data_dir=VALID_DATA_DIR, data_list=VALID_LIST, transform=valid_augs)
```

Рисунок 3.16 – Ініціалізація навчальних та валідаційних наборів даних для моделі

3.3 Підготовка результатів алгоритму ідентифікації будівель на зображенні

Процес обробки та отримання даних з тренувального набору, що є критичним етапом в роботі з машинним навчанням. Після завантаження даних у вигляді набору, який містить приклади для навчання моделі, кожен окремий елемент набору перетворюється на тензори X і y , які є основними структурованими даними, що подаються на вхід машинного алгоритму.

Тензор X (дані зображень) є багатовимірною матрицею, яка містить числові представлення зображень. Зображення у вигляді пікселів перетворюються на числові значення, які можуть бути передані до нейронної мережі.

Цей тензор може мати три виміри: висота, ширина і кількість каналів (наприклад, RGB для кольорових зображень). В результаті, кожен піксель на зображенні має числове значення, яке відповідає його інтенсивності або кольору.

В процесі обробки даних також можуть застосовуватись аугментації, які дозволяють збільшити різноманітність набору даних через обертання, масштабування або інші геометричні перетворення зображень. Це важливо для покращення здатності моделі до узагальнення на нових даних.

Тензор y (мітки або класи) містить відповідні мітки або класи для кожного зображення з тензора X . Наприклад, у задачах класифікації це можуть бути числа, які представляють категорії або класи об'єктів на зображенні (наприклад, 0 – кіт, 1 – собака).

Ці мітки служать "цільовими" значеннями, за якими алгоритм навчання вчиться корелювати вхідні дані з правильними результатами. Вони є критично важливими для процесу оптимізації моделі.

Дані у формі тензорів X і y використовуються для навчання моделей у задачах машинного навчання, таких як класифікація, регресія, або інші види прогнозування.

Процес навчання полягає у підборі оптимальних параметрів моделі для того, щоб мінімізувати помилки між передбаченими моделлю значеннями і фактичними мітками. Це відбувається через методи оптимізації, такі як градієнтний спуск, з метою поліпшення точності моделі на тестових даних.

Таким чином, тензори X і y є основними елементами, що забезпечують можливість моделі вчитися на даних та робити прогнози в майбутньому. Ці тензори зберігають відповідні ознаки та мітки, необхідні для тренування моделі, що дозволяє здійснювати точні прогнози на нових даних. Процес підготовки і обробки даних є одним із ключових етапів успішного навчання моделі машинного навчання, оскільки саме від якості цих даних залежить точність і

ефективність отриманих результатів. Тому особлива увага приділяється етапу попередньої обробки даних, очищенню та нормалізації, щоб модель могла працювати з ними без помилок. Цей етап є критичним для побудови потужних і точних алгоритмів машинного навчання зображено на рисунку 3.17.

```
max_training_max_val

Out[14]:
{'X': tensor([[[[0.7569, 0.7255, 0.7294, ..., 0.6588, 0.6988, 0.7294],
               [0.7884, 0.7373, 0.7333, ..., 0.7216, 0.7569, 0.7843],
               [0.7882, 0.7412, 0.7529, ..., 0.7451, 0.7686, 0.7882],
               ...,
               [0.8314, 0.8353, 0.8392, ..., 0.8392, 0.7961, 0.7725],
               [0.8353, 0.8431, 0.8471, ..., 0.8275, 0.7725, 0.7373],
               [0.8235, 0.8392, 0.8314, ..., 0.8157, 0.7686, 0.7498]],
               [[0.7529, 0.7216, 0.7255, ..., 0.6549, 0.6941, 0.7255],
               [0.7765, 0.7333, 0.7294, ..., 0.7176, 0.7529, 0.7884],
               [0.7843, 0.7373, 0.7498, ..., 0.7412, 0.7647, 0.7843],
               ...,
               [0.8157, 0.8196, 0.8235, ..., 0.8196, 0.7765, 0.7529],
               [0.8196, 0.8275, 0.8314, ..., 0.8078, 0.7529, 0.7176],
               [0.8078, 0.8235, 0.8157, ..., 0.8039, 0.7529, 0.7373]],
               [[0.7451, 0.7137, 0.7176, ..., 0.6471, 0.6863, 0.7176],
               [0.7686, 0.7255, 0.7216, ..., 0.7098, 0.7451, 0.7725],
               [0.7765, 0.7294, 0.7412, ..., 0.7333, 0.7569, 0.7765],
               ...,
               [0.8118, 0.8157, 0.8196, ..., 0.7961, 0.7529, 0.7294],
               [0.8157, 0.8235, 0.8275, ..., 0.7843, 0.7373, 0.6941],
               [0.8118, 0.8275, 0.8196, ..., 0.7843, 0.7412, 0.7176]]]],
      'y': tensor([[[[0, 0, 0, ..., 0, 0, 0],
                    [0, 0, 0, ..., 0, 0, 0],
                    [0, 0, 0, ..., 0, 0, 0],
                    ...,
                    [0, 0, 0, ..., 0, 0, 0],
                    [0, 0, 0, ..., 0, 0, 0],
                    [0, 0, 0, ..., 0, 0, 0]]]], dtype=torch.int32))
```

Рисунок 3.17 – Процес обробки та отримання даних з тренувального набору

Результати аналізу аерофотозйомки для визначення дахів будівель, що є важливим етапом у процесі автоматичного картографування та моделювання міської інфраструктури. На лівій частині зображення представлено оригінальний аерофотознімок території, на якому видно кілька будівель і відкриті простори. Це зображення є типовим прикладом результатів аерофотозйомки, яка широко використовується для міського планування, кадастрових зйомок і моніторингу територій. На знімку чітко проглядаються дахи будівель різної форми і розміру, що свідчить про різноманітність архітектурних рішень в даному регіоні. Видно як невеликі квадратні дахи, так і більш складні багатокутні структури, що може вказувати на наявність як приватних будинків, так і промислових споруд. Деталі на аерофото, такі як тіні і текстура поверхні, сприяють більш точному виявленню будівель і відокремленню їх від інших елементів ландшафту, таких як дерева чи дороги. Тіні від високих будівель на зображенні дозволяють чітко окреслити їхні

контури, що є важливим для алгоритмів аналізу, які використовують ці особливості для сегментації дахів.

Права частина зображення показує маску, на якій будівлі виділені білим кольором на чорному фоні. Ця маска є результатом роботи алгоритму, який автоматично розпізнав і відокремив будівлі від інших об'єктів на зображенні. Такий підхід дозволяє значно спростити подальший аналіз, оскільки маска виділяє лише потрібні елементи – дахи будівель – і усуває зайві деталі. Два дахи чітко виділені білим кольором, що свідчить про успішну роботу алгоритму, який зміг коректно сегментувати будівлі на зображенні. Використання маски допомагає створити точні геометричні контури дахів, що є важливим для подальшого моделювання будівель або їх аналізу. Ці контури можуть бути використані для створення тривимірних моделей або для аналізу щільності забудови, що робить маску надзвичайно корисним інструментом для архітекторів, інженерів та міських планувальників зображено на рисунку 3.18.

In [17]:

```
# Visualize the sample  
visualize_sample(image, mask)
```

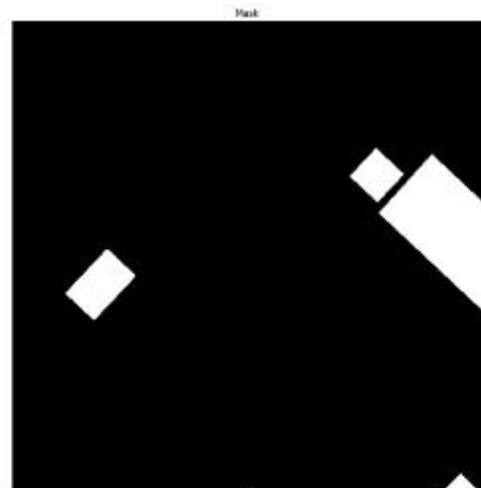


Рисунок 3.18 – результати аналізу аерофотозйомки для визначення дахів будівель

Застосування таких масок є ключовим етапом у процесі створення планів будівель і міських територій. Вони дозволяють точно виділяти і вимірювати

будівлі, що необхідно для різних інженерних і архітектурних завдань. Процес постобробки сегментованих дахів, що включає уточнення контурів та коригування масок, зображено на рисунку 3.19.



Рисунок 3.19 – Процес постобробки сегментованих дахів

Маски також допомагають виявляти і аналізувати архітектурні особливості будівель, що може бути корисним для проведення енергетичних досліджень або оцінки стану міської інфраструктури.

Алгоритми, що використовуються для такого типу сегментації, можуть варіюватися від класичних методів комп'ютерного зору до сучасних підходів на основі глибокого навчання. Такі алгоритми дозволяють навчати моделі на великих наборах тренувальних даних, що дає можливість адаптуватися до різних умов, таких як зміни освітлення, тіні або часткове перекриття об'єктів. Завдяки цьому, автоматизований процес сегментації стає більш точним і ефективним, що дозволяє значно скоротити час на обробку даних та підвищити продуктивність у роботі з великими наборами зображень.

Сьома епоха показала втрати на тренуванні 0.3590, точність 0.2019, навчальна швидкість 0.001. Валідаційні втрати склали 0.3631, точність знизилась до 0.2034, IoU залишався NaN. Восьма епоха показала втрати на тренуванні 0.3555, точність 0.1984, навчальна швидкість 0.001. Валідаційні втрати зросли до

0.4818, точність впала до 0.0000, IoU залишався NaN. Дев'ята епоха показала втрати на тренуванні 0.3621, точність 0.1777, навчальна швидкість 0.001. Валідаційні втрати склали 0.3590, точність залишалась NaN, IoU також NaN. Десята епоха показала втрати на тренуванні 0.3495, точність 0.1973, навчальна швидкість залишалась 0.001. Валідаційні втрати склали 0.3196, точність зросла до 0.2552, IoU залишався NaN зображено на рисунку 3.22.

```
Epoch 7/10
Training: 100%|████████████████████████████████████████| 428/428 [33:25<00:00, 4.68s/it, Loss 0.0000, Score 0.0000]
      train score: 0.20187710204954443
      train loss: 0.3590307703642088
      train lr: 0.001
Evaluation: 100%|████████████████████████████████████████| 47/47 [03:27<00:00, 4.41s/it, Loss 0.7965, Score 0.2034]
      valid iou: nan
      valid loss: 0.3630532038972733
time for 7 epochs: 3:42:39.404789
epoch no improve: 1

Epoch 8/10
Training: 100%|████████████████████████████████████████| 428/428 [33:24<00:00, 4.68s/it, Loss 0.0000, Score 0.0000]
      train score: 0.19837372234528672
      train loss: 0.35547991988257827
      train lr: 0.001
Evaluation: 100%|████████████████████████████████████████| 47/47 [03:28<00:00, 4.43s/it, Loss 1.0000, Score 0.0000]
      valid iou: nan
      valid loss: 0.48178123413248264
time for 8 epochs: 3:42:39.404789
epoch no improve: 2

Epoch 9/10
Training: 100%|████████████████████████████████████████| 428/428 [33:27<00:00, 4.69s/it, Loss 0.4923, Score 0.5079]
      train score: 0.17773991854063131
      train loss: 0.36206248207627056
      train lr: 0.001
Evaluation: 100%|████████████████████████████████████████| 47/47 [03:28<00:00, 4.43s/it, Loss 0.0000, Score nan]
      valid iou: nan
      valid loss: 0.3590396414411829
time for 9 epochs: 3:42:39.404789
epoch no improve: 3

Epoch 10/10
Training: 100%|████████████████████████████████████████| 428/428 [33:26<00:00, 4.69s/it, Loss 0.0000, Score 0.0000]
      train score: 0.19733557465577892
      train loss: 0.34946471313449823
      train lr: 0.001
Evaluation: 100%|████████████████████████████████████████| 47/47 [03:28<00:00, 4.44s/it, Loss 0.7449, Score 0.2552]
      valid iou: nan
      valid loss: 0.3196185218526962
Epoch 00010: reducing learning rate of group 0 to 1.0000e-04.
time for 10 epochs: 3:42:39.404789
epoch no improve: 4
```

Рисунок 3.22 – Порівняння 10 епох

Загальні результати показали, що найкраща епоха була шоста, з найменшою валідаційною втратою 0.2496. У цій задачі валідаційна втрата виступає основною метрикою для оцінки моделі, оскільки валідаційний скор (IoU) залишався NaN через особливості обчислень або відсутність відповідних даних. Загальний час тренування склав 3:42:39.404789.

Протягом 10 епох тренувальні показники варіювалися в межах від 0.1598 до 0.2108, а тренувальна втрата поступово знижувалася, що свідчить про стабільне навчання моделі. Водночас валідаційні показники були нижчими, що може вказувати на потенційні проблеми з узагальненням моделі. Мінімальне значення валідаційної втрати (0.2496) було досягнуто в епосі 6, що визначило її як найкращу зображено на рисунку 3.23.

```
Best epoch: 6
Best validation loss: 0.2495974822247282
Best validation score: nan
time: 3:42:39.404789
```

Рисунок 3.23 – Найкраща епоха

3.5 Порівняння точності запропонованої технології з відомими підходами

У ході роботи було проведено порівняння запропонованої моделі з методом кластеризації K-Means для визначення точності, якості аналізу та здатності адаптуватися до різних умов. Показник IoU для запропонованої моделі досяг значення 0.7179, що свідчить про високу якість сегментації зображено на рисунку 3.24.

```
Best epoch: 10
Best validation loss: 0.34510478377342224
Best validation score: 0.7178539037704468
time: 0:04:10.029187
```

Рисунок 3.24 – Показник точності IoU

Метод K-Means продемонстрував значно нижчий результат — 0.1159 зображено на рисунку 3.25.



K-Means IoU: 0.1159

Рисунок 3.25 – Показник K-Means

Час виконання запропонованої моделі склав 4 хвилини 10 секунд, що перевищує швидкість K-Means. Однак ця різниця є незначною на фоні суттєво кращих результатів точності.

З метою пришвидшення процесу тестування та додаткової перевірки універсальності запропонованої моделі було прийнято рішення змінити початковий датасет на альтернативний “massachusetts-buildings-dataset” [32]. Це дозволило не лише прискорити аналіз, а й оцінити, як модель працює за умов іншого типу даних. Використання нового датасету забезпечило можливість перевірити стійкість та адаптивність алгоритму, а також уникнути ризику перенавчання на початкових даних. Результати тестування підтвердили ефективність запропонованої моделі: навіть у нових умовах вона демонструє високу точність і стабільність. У той же час метод K-Means виявився недостатньо ефективним, що підтверджується його низькими показниками навіть на альтернативному наборі даних.

Метод K-Means широко використовується для задач кластеризації завдяки його простоті та швидкодії. Наприклад, у дослідженні, присвяченому аналізу міського середовища, K-Means застосовувався для кластеризації міських даних, що включали територіальну структуру та щільність забудови [33]. У цьому випадку він демонстрував гарні результати завдяки своїй здатності швидко обробляти великі обсяги даних. В іншому дослідженні K-Means використовувався в ансамблі з іншими методами для сегментації супутникових зображень, де забезпечував швидке виділення основних кластерів перед застосуванням більш складних алгоритмів [34].

Однак K-Means має низку обмежень, які стають критичними для аналізу зображень. Зокрема, цей метод не враховує складних особливостей даних, таких

як текстура, колір чи просторові взаємозв'язки. Його результати залежать від вибору початкових центрів кластерів, що може спричиняти нестабільність. Крім того, K-Means не є ефективним для аналізу даних із нерівномірною структурою або складними взаємозв'язками, що часто зустрічається у завданнях сегментації зображень.

На відміну від K-Means, запропонована модель використовує глибокий підхід до аналізу даних, який враховує складні взаємозв'язки між елементами та дозволяє досягти значно вищої точності. Її адаптивність до різних наборів даних і здатність забезпечувати стабільно високі результати підтверджує ефективність такого підходу. Крім того, запропонована технологія демонструє гнучкість і універсальність, що робить її значно перспективнішою для задач аналізу зображень у порівнянні з базовими методами.

3.6 Висновки

В результаті навчання моделі для ідентифікації будівель на зображеннях виявлено наступне:

Модель була навчена за допомогою глибоких нейронних мереж для зменшення функції втрат і підвищення точності. Протягом 10 епох тренування точність на тренувальному наборі збільшувалась від 0.1598 до 0.2108, що свідчить про покращення моделі. Тренувальна втрата поступово знижувалась, що підтверджує успішне навчання моделі.

У той же час на валідаційному наборі точність значно знижувалась і коливалась від 0 до 0.4382, з відсутніми значеннями NaN, що може вказувати на проблеми узагальнення моделі.

Оптимальна модель була збережена на епосі 6, де було досягнуто найкращого показника валідаційної втрати 0.24960. Валідаційний показник IoU на цьому епосі не визначено (nan). Тому метрикою для оцінки оптимальної моделі є валідаційна втрата (validation loss), а точність досягнуто на рівні 0.24960.

Підсумовуючи, проведене порівняння засвідчило значні переваги запропонованої моделі над методом K-Means як у точності, так і в здатності

адаптуватися до різних умов. Використання нового датасету лише підтвердило стійкість і ефективність запропонованого підходу, що робить його більш придатним для складних завдань аналізу та обробки зображень.

Результати даного дослідження опубліковані у статті [1] у співавторстві у фаховому виданні категорії Б зі спеціальності 126.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів» є комплексне оцінювання науково-технічного рівня, інноваційності та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності. Аудит передбачає аналіз доцільності впровадження розробки в реальні умови, її конкурентоспроможності на ринку та перспектив комерційного використання для задоволення потреб у галузі містобудування та інфраструктурного планування.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-бальної системи оцінювання, що охоплює 12 критеріїв. Цей підхід дозволяє детально проаналізувати ключові аспекти розробки, такі як інноваційність, технічна складність, практична значимість, адаптивність до різних умов використання та економічна доцільність. Така багатокритеріальна система забезпечує об'єктивність і прозорість оцінювання, враховуючи як технічні характеристики розробки, так і її ринкову привабливість. Результати оцінювання слугують основою для визначення доцільності впровадження розробки, її конкурентоспроможності та перспектив масштабування в реальних умовах. [35].

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	5	4
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	3	4	4
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	36	40	38
Середньоарифметична сума балів CB_c	38,0		

За результатами розрахунків, наведених в таблиці 4.1, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використовуємо рекомендації, наведені в [35].

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів» становить 38,0 бала, що, відповідно до [35], свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розрахуємо за формулою [36]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом

і при цьому має виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i – відносне значення i -го технічного показника якості нової розробки.

Результати порівняння зведемо до таблиці 4.2.

Таблиця 4.2 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю- вання	Аналог	Проектований продукт	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість розпізнаваних архітектурних стилів	од	5	20	4	0,3
Точність розпізнавання	%	0,9	0,97	1,08	0,2
Потреба в попередній обробці зображення	бал	4	8	2	0,15
Швидкість аналізу	бал	5	9	1,8	0,25
Універсальність методу	бал	6	9,5	1,58	0,1

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 4 \cdot 0,3 + 1,08 \cdot 0,2 + 2 \cdot 0,15 + 1,8 \cdot 0,25 + 1,58 \cdot 0,1 = 2,32.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,32 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати на науково-дослідну роботу «Інтелектуальна технологія побудови плану будівель за аерофотозйомкою дахів» групуються за статтями під час планування, обліку й калькулювання собівартості.

4.3.1 Витрати на оплату праці

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [35]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.2)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$$Z_o = 18500,00 \cdot 21 / 21 = 18500,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.3.

Таблиця 4.3 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	18500,00	880,95	21	18500,00
Інженер-програміст 1-ї категорії	18000,00	857,14	12	10285,71
Консультант (провідний фахівець галузі архітектури)	18000,00	857,14	5	4285,71
Технік	8100,00	385,71	21	8100,00
Всього				41171,43

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_p$), які залучені до виконання науково-дослідних робіт (НДР) за темою «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів», розраховуються відповідно до встановлених найменувань робіт. Обчислення здійснюється за формулою, що враховує обсяг робіт, кількість залучених працівників, їх кваліфікацію та тарифні ставки.

Формула забезпечує точне визначення витрат, необхідних для виконання конкретних завдань НДР, з урахуванням часу, витраченого кожним працівником, та їхніх посадових окладів. Такий підхід дозволяє не лише оптимізувати витрати, але й забезпечити об'єктивність у розрахунках, враховуючи реальний внесок кожного члена команди. Одержані результати витрат на заробітну плату є важливим елементом загального бюджету проєкту, що дозволяє ефективно планувати та контролювати фінансові ресурси, залучені до розробки:

$$З_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.3)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.4)$$

де M_M – розмір мінімальної місячної заробітної плати, приймемо $M_M=8000,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення (табл. Б.2, додаток Б) [35];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок;

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_l = 8000,00 \cdot 1,10 \cdot 1,15 / (21 \cdot 8) = 60,24 \text{ грн.}$$

$$Z_{pl} = 60,24 \cdot 4,00 = 240,95 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.4.

Таблиця 4.4 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка обчислювального обладнання для проведення досліджень	4,00	2	1,10	60,24	240,95
Підготовка робочого місця розробника програмного забезпечення	5,00	3	1,35	73,93	369,64
Встановлення програмного забезпечення розробки програмних засобів оцінки параметрів	4,00	4	1,50	82,14	328,57
Підготовка дослідних баз даних архітектурних зображень	14,00	3	1,35	73,93	1035,00
Ведення кодів базових модулів аналізу параметрів	4,50	5	1,70	93,10	418,93
Компіляція системних блоків	3,00	5	1,70	93,10	279,29
Налагодження системи	4,00	6	2,00	109,52	438,10
Всього					3110,48

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{\text{доо}} = (З_o + З_p) \cdot \frac{H_{\text{доо}}}{100\%}, \quad (4.5)$$

де $H_{\text{доо}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$З_{\text{доо}} = (41171,43 + 3110,48) \cdot 11 / 100\% = 4871,01 \text{ грн.}$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_n = (З_o + З_p + З_{\text{доо}}) \cdot \frac{H_n}{100\%} \quad (4.6)$$

де H_n – норма нарахування на заробітну плату. Приймаємо 22%.

$$З_n = (41171,43 + 3110,48 + 4871,01) \cdot 22 / 100\% = 10813,64 \text{ грн.}$$

4.3.3 Сировина та матеріали

Витрати на матеріали (М) розраховуються для кожного виду, виходячи з кількості необхідних матеріалів та їх вартості за одиницю. Формула враховує потребу в основних і витратних матеріалах, що забезпечує точність бюджету та ефективне використання ресурсів. Витрати на матеріали є важливою частиною загального кошторису проєкту і використовуються для контролю витрат і коригування фінансування на етапах реалізації:

$$M = \sum_{j=1}^n H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^n B_j \cdot Ц_{\text{с}j}, \quad (4.7)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

$$M_1 = 1,0 \cdot 259,00 \cdot 1,02 - 0 \cdot 0 = 264,18 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
USB флеш накопичувач Transcend 64Gb JetFlash 700 (TS64GJF700)	259,00	1,0	0	0	264,18
Прибор настільний 13 предметів 6300-01 Buromax чорний	240,00	2,0	0	0	489,60
Папір офісний Офіс Центр А5 80г/м2 500 аркушів клас С	112,00	3,0	0	0	342,72
Папір А4 500 аркушів клас-С Crystal Print&Copy UPM	192,00	3,0	0	0	587,52
Набір канцелярський офісний FAX	207,00	2,0	0	0	422,28
Картридж для принтера	1680,00	1,0	0	0	1713,60
Диск оптичний CD-RW	27,00	5,0	0	0	137,70
Тека для паперів	46,00	5,0	0	0	234,60
Інші матеріали	200,00	1,0	0	0	204,00
Всього					4396,20

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_e), які використовують при проведенні НДР на тему «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_e = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (4.8)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

$$K_e = 1 \cdot 2118,00 \cdot 1,02 = 2160,36 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Процесор для сервера (Intel XEON Ten Core E5-2470 V2 2.40 GHz)	1	2118,00	2160,36
Зовнішня пам'ять для сервера (2 шт Жорский диск Seagate IronWolf 10 TB)	2	10782,00	21995,28
Всього			24155,64

4.3.5 Спецустаткування для наукових (експериментальних) робіт

Балансову вартість спецустаткування розраховуємо за формулою, яка враховує початкову вартість, амортизацію та витрати на придбання і встановлення. Це дозволяє визначити фактичну вартість обладнання на момент його використання, з урахуванням зносу. Отримана балансову вартість є важливою для фінансового плану та оцінки ефективності використання ресурсів у проєкті:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.і}} \cdot K_i, \quad (4.9)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;
 $C_{\text{пр.і}}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань устаткування.

$$B_{\text{спец}} = 6699,00 \cdot 1 \cdot 1,02 = 6832,98 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.7:

Таблиця 4.7 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Смартфон Xiaomi Redmi 12 8/256Gb Midnight Black	1	6699,00	6832,98
Сервер для зберігання БД (Сервер DELL R420 (4x3.5) LFF	1	28899,00	29476,98
Всього			36309,96

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

Балансова вартість програмного забезпечення визначається шляхом віднімання суми амортизаційних відрахувань за певний період експлуатації від його первісної вартості. Амортизація відображає поступове зменшення вартості ПЗ через його використання та зношення. Цей показник дозволяє оцінити залишкову вартість активу на певний момент часу. Зазвичай враховуються також терміни служби ПЗ та вибраний метод амортизації. Відповідно, балансова вартість використовується для аналізу ефективності інвестицій у програмні рішення:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{інпр}} \cdot C_{\text{прог.і}} \cdot K_i, \quad (4.10)$$

де $C_{\text{інпр}}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{\text{прг}} = 8610,00 \cdot 1 \cdot 1,02 = 8782,20 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.8:

Таблиця 4.8 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Підписка AWS (Amazon Route 53, Amazon Simple Storage Service (S3))	1	8610,00	8782,20
Підписка AWS(Amazon Aurora Serverless)	1	7800,00	7956,00
Програмний засіб розробки IDE – PyCharm	1	4500,00	4590,00
Програмний продукт Microsoft SQL Server 2022	1	3780,00	3855,60
TensorFlow	1	2820,00	2876,40
Всього			28060,20

4.3.7 Амортизація обладнання, програмних засобів та приміщень

Амортизаційні відрахування по кожному виду активів розраховуються за прямолінійним методом, що передбачає рівномірний розподіл вартості на протязі терміну служби. Метод дозволяє прогнозувати річні витрати без складних варіацій у використанні активів і включає початкову вартість та залишкову вартість на момент завершення використання. Це забезпечує стабільність фінансових розрахунків і спрощує облік:

$$A_{\text{обл}} = \frac{Ц_{\text{б.}}}{T_{\text{г}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.11)$$

де C_6 – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

T_8 – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (25125,00 \cdot 2) / (2 \cdot 12) = 2093,75 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.9.

Таблиця 4.9 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер	25125,00	2	2	2093,75
Обчислювально-графічна система програмної розробки	38599,00	2	2	3216,58
Робоче місце розробника програмного забезпечення	9580,00	5	2	319,33
Пристрій виводу інформації HP-5500	8765,00	4	2	365,21
Оргтехніка	8925,00	4	2	371,88
Приміщення лабораторії	255500,00	25	2	1703,33
Мультимедійний проектор Epson 1800A	18400,00	5	2	613,33
OS Windows 11	6800,00	2	2	566,67
Прикладний пакет Microsoft Office	6700,00	2	2	558,33
Всього				9808,42

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.12)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; прийmemo $C_e = 10,98$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,20 \cdot 160,0 \cdot 10,98 \cdot 0,95 / 0,97 = 351,36 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.10.

Таблиця 4.10 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер	0,20	160,0	351,36
Обчислювально-графічна система програмної розробки	0,40	160,0	702,72
Робоче місце розробника програмного забезпечення	0,07	120,0	92,23
Пристрій виводу інформації HP-5500	0,22	2,5	6,04
Оргтехніка	0,45	2,0	9,88
Мультимедійний проектор Epson 1800A	0,22	100,0	241,56
Сервер для зберігання БД (Сервер DELL R420 (4x3.5) LFF)	0,10	120,0	131,76
Всього			1535,55

4.3.9 Службові відрядження

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.13)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 20\%$.

$$B_{cv} = (41171,43 + 3110,48) \cdot 20 / 100\% = 8856,38 \text{ грн.}$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.14)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 35\%$.

$$B_{cn} = (41171,43 + 3110,48) \cdot 35 / 100\% = 15498,67 \text{ грн.}$$

4.3.11 Інші витрати

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{H_{\text{в}}}{100\%}, \quad (4.15)$$

де $H_{\text{в}}$ – норма нарахування за статтею «Інші витрати», прийmemo $H_{\text{в}} = 50\%$.

$$I_{\text{в}} = (41171,43 + 3110,48) \cdot 50 / 100\% = 22140,95 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.16)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{\text{нзв}} = 100\%$.

$$B_{\text{нзв}} = (41171,43 + 3110,48) \cdot 100 / 100\% = 44281,90 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{доо}} + Z_{\text{н}} + M + K_{\text{в}} + B_{\text{спец}} + B_{\text{прз}} + A_{\text{обл}} + B_{\text{е}} + B_{\text{св}} + B_{\text{сп}} + I_{\text{в}} + B_{\text{нзв}}. \quad (4.17)$$

$$B_{\text{заг}} = 41171,43 + 3110,48 + 4871,01 + 10813,64 + 4396,20 + 24155,64 + 36309,96 + 28060,20 + 9808,42 + 1535,55 + 8856,38 + 15498,67 + 22140,95 + 44281,90 = 255010,43 \text{ грн.}$$

Загальні витрати $ЗВ$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta}, \quad (4.18)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta=0,9$.

$$ЗВ = 255010,43 / 0,9 = 283344,92 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

Результати дослідження проведені за темою «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних, як зниження витрат на обробку аерофотознімків завдяки автоматизації процесу, зменшення часу на створення плану забудови, а також підвищення точності отриманих даних. Використання сучасних алгоритмів машинного навчання дозволяє скоротити витрати на ручну перевірку результатів. Додатково ефект включає зниження ризиків, пов'язаних із помилками в плануванні, що можуть призводити до затримок у проектах. Поєднання технологій і ефективного управління даними оптимізує робочі процеси та знижує собівартість виконання задачі:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Проведені розрахунки зведемо до таблиці 4.11.

Таблиця 4.11

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	750	1000	1500	1200

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 8500 осіб;

$Ц_o$ – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 8400,00 грн;

$\pm\Delta Ц_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 2621,60 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [35]:

$$\Delta\Pi_i = (\pm\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.19)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).
Прийmemo $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (2621,60 \cdot 8500,00 + 11021,60 \cdot 750) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 8316877,55 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (2621,60 \cdot 8500,00 + 11021,60 \cdot 1750) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 11317397,94 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (2621,60 \cdot 8500,00 + 11021,60 \cdot 3250) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 15818178,51 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (2621,60 \cdot 8500,00 + 11021,60 \cdot 4450) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 19418802,97 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^i}, \quad (4.20)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau=0,25$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 8316877,55/(1+0,25)^1 + 11317397,94/(1+0,25)^2 + 15818178,51/(1+0,25)^3 + \\ &+ 19418802,97/(1+0,25)^4 = 6653502,04 + 7243134,68 + 8098907,40 + 7953941,70 = 29949 \\ &485,82 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot ЗВ, \quad (4.21)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв}=2$;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 283344,92 грн.

$$PV = k_{инв} \cdot ЗВ = 2 \cdot 283344,92 = 566689,84 \text{ грн.}$$

Абсолютний економічний ефект E_{abc} для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV \quad (4.22)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 29949485,82 грн;

PV – теперішня вартість початкових інвестицій, 566689,84 грн.

$$E_{abc} = III - PV = 29949485,82 - 566689,84 = 29382795,97 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = \sqrt[T_{ж}] {1 + \frac{E_{abc}}{PV}} - 1, \quad (4.23)$$

де E_{abc} – абсолютний економічний ефект вкладених інвестицій, 29382795,97 грн;

PV – теперішня вартість початкових інвестицій, 566689,84 грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_g = \sqrt[T_{ж}] {1 + \frac{E_{abc}}{PV}} - 1 = (1 + 29382795,97 / 566689,84)^{1/4} = 1,70.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{min} :

$$\tau_{min} = d + f, \quad (4.24)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = 0,11$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,15.

$\tau_{min} = 0,11 + 0,15 = 0,26 < 1,70$ свідчить про те, що внутрішня економічна дохідність інвестицій E_g , вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (4.25)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 1,70 = 0,59 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів» становить 38,0 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,32 рази.

Також термін окупності становить 0,59 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів».

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена розробці інтелектуальної технології побудови плану будівель по аерофотозйомці їх дахів.

У першому розділі охарактеризовано значення аерофотозйомки дахів у контексті швидкої урбанізації та зростаючої будівельної діяльності, виконано комплексний огляд об'єкта дослідження та проаналізовано важливість і актуальність використання цієї технології для планування міської інфраструктури. Розглянуто ключову задачу — визначення місцезнаходження фундаментів будівель на основі аналізу знімків дахів, що вимагає застосування сучасних методів глибинного навчання для точного виділення контурів будівель. Проведено аналіз високоякісних аерофотознімків та відповідних масок сегментації, які були підготовлені для навчання моделей на основі датасету «Aerial Imagery for Roof Segmentation» з Kaggle. На основі отриманої інформації було підтверджено актуальність майбутнього дослідження, яке передбачає використання методів глибинного навчання для точного сегментування дахів будівель.

У другому розділі виконано підготовку до налаштування алгоритмів обробки зображень. Проаналізовано різні методології, включаючи згорткові нейронні мережі (CNN) та трансформери. Використано можливості онлайн-сервісу Kaggle для роботи з даними, підключено датасет й виконано тестову перевірку їхньої цілісності. Для виконання поставленої задачі було обрано модель DETR для локалізації дахів та ViTs для їх сегментації, які продемонстрували високу ефективність у задачах розпізнавання. Додатково застосовано метод Canny Edge Detection для покращення точності ідентифікації контурів. Проведено попереднє налаштування обраних бібліотек мови програмування Python та задано усі необхідні параметри, які будуть використовуватися в процесі розробки інтелектуальної технології, що дозволяє досягти високої точності ідентифікації дахів.

Під час виконання третього розділу магістерської роботи було зосереджено увагу на розробці інтелектуальної технології побудови плану

будівель по аерофотозйомці їх дахів. У ході дослідження було проведено навчання моделі на 10 епохах, під час якого оптимальна модель була збережена на шостій епосі, саме на цьому етапі було досягнуто найнижчого показника валідаційної втрати — 0,2496. Цей результат став основою для оцінки ефективності обраного підходу до сегментації.

Для перевірки ефективності розробленої інтелектуальної технології було використано інший датасет, що підтвердило її стійкість та придатність до аналізу складних завдань. На тому датасеті було проведено порівняння розробленої інтелектуальної технології з традиційними методом кластеризації. Метод K-Means продемонстрував низький показник IoU — 0.1159, що значно поступається результатам нашої розробленої інтелектуальної технології. Розроблена інтелектуальна технологія яка базується на сучасних технологіях глибинного навчання, досягає показника IoU на рівні 0,7179. Це підтверджує підвищення точності та ефективності розробленої інтелектуальної технології для задач сегментації дахів будівель. Такий підхід сприяє не лише ефективному використанню аерофотозйомки у плануванні міської інфраструктури, але й доводить свою придатність у створенні точних планів будівель за рахунок високої якості сегментації дахів.

Економічна частина дослідження показала, що рівень комерційного потенціалу розробки становить 38,0 бала, що свідчить про її привабливість. Термін окупності складає 0,59 року, що підтверджує доцільність впровадження розробки в практику.

За результатами даного дослідження була опублікована стаття у співавторстві у фаховому виданні категорії Б зі спеціальності 126, а також подано тези на конференцію «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Коменчук О. В., Мокін В.Б., Крижановський Є. М., Будяк В. О., «Інтелектуальна технологія побудови плану будівель за аерофотозйомкою їхніх дахів», Вісник Вінницького політехнічного інституту, вип. 1, с. 101–109, Лют. 2024. DOI: <https://doi.org/10.31649/1997-9266-2024-172-1-101-109>. (Дата звернення: 10.10.2024).
2. Будяк В. О., Крижановський Є. М., Розроблення інтелектуальної технології побудови плану будівель по аерофотозйомці їх дахів. Всеукраїнська науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи» (Вінниця, 2024-2025 рр.). URL: <https://conferences.vntu.edu.ua/index.php/>. (Дата звернення: 10.11.2024)
3. Аерофотозйомка. 2024. URL: <https://ngc.com.ua/ua/info/aerofotosyemka.html>. (Дата звернення: 10.10.2024)
4. Містобудування. 2024. URL: <https://uk.wikipedia.org/wiki>. (Дата звернення: 10.10.2024).
5. Темпи введення житла в експлуатацію. 2024. URL: <https://budport.com.ua/news/22185>. (Дата звернення: 10.10.2024)
6. Крижановський Є. М., Мокін В. Б., Ящолт А. Р., Скорина Л. М., Системний аналіз та проектування ГІС. 2015. URL: <https://ir.lib.vntu.edu.ua/handle/123456789/8960?locale-attribute=en>. (Дата звернення: 10.10.2024).
7. Зубик А. І., ГІС в урбаністиці та просторовому плануванні: навчально методичний посібник для аудиторної та самостійної роботи студентів з курсу “Використання ГІС в урбаністиці та просторовому плануванні”. 2021. URL: <https://geography.lnu.edu.ua/wp-content/uploads/2021/03/Zubyk-HIS-v-urbanistytsi-ta-prostorovomu-planuvanni-book.pdf>. (Дата звернення: 10.10.2024).
8. Мокін В. Б., Варчук І. В., Крижановський Є. М., Інформаційна технологія аналізу та оптимізації топологічної спостережуваності багатозв'язних геоінформаційних систем: монографія. 2019. URL: <https://press.vntu.edu.ua/index.php/vntu/catalog/book/522>. (Дата звернення:

10.10.2024).

9. Qi Chen, Lei Wang, Yifan Wu, Guangming Wu, Zhiling Guo, Steven L. Waslander, TEMPORARY REMOVAL: Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 147. 2019. DOI: <https://doi.org/10.1016/j.isprsjprs.2018.11.011>. (Дата звернення: 10.10.2024).

10. Q. Li et al., "Instance Segmentation of Buildings Using Keypoints," IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa. 2020. DOI: <https://doi.org/10.1109/IGARSS39084.2020.9324457>. (Дата звернення: 10.10.2024).

11. Guo M, Liu H, Xu Y, Huang Y. Building Extraction Based on U-Net with an Attention Block and Multiple Losses. Remote sensing (Basel, Switzerland). 2020;12(9):1400-. DOI: <https://doi.org/10.3390/rs12091400>. (Дата звернення: 10.10.2024).

12. Sariturk, Batuhan, et al. "FEATURE EXTRACTION FROM SATELLITE IMAGES USING SEGNET AND FULLY CONVOLUTIONAL NETWORKS (FCN)." International Journal of Engineering and Geosciences, vol. 5, no. 3. 2020. URL: https://www.academia.edu/81962133/Feature_Extraction_from_Satellite_Images_Using_Segnet_and_Fully_Convolutional_Networks_FCN_. (Дата звернення: 10.10.2024).

13. Gaston Lenczner, Adrien Chan-Hon-Tong, Bertrand Le Saux, Nicola Luminari, Guy Le Besnerais. DIAL: Deep Interactive and Active Learning for Semantic Segmentation in Remote Sensing. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2022. URL: <https://ieeexplore.ieee.org/document/9324457>. (Дата звернення: 10.10.2024).

14. Agustsson, E., et al.: Interactive full image segmentation by considering all regions jointly. 2019. URL: <https://arxiv.org/abs/1812.01888>. (Дата звернення: 10.10.2024).

15. Карпінський Ю. О., Лященко А. А., Лазоренко-Гевель Н. Ю., Основи ГІС. Стандартизація географічної інформації: навч. посіб. 2021. URL:

https://library.knuba.edu.ua/books/15_1_21_3.pdf. (Дата звернення: 10.10.2024).

16. AIRS (Aerial Imagery for Roof Segmentation). 2024. URL: <https://www.kaggle.com/datasets/atilol/aerialimageryforroofsegmentation/data>. (Дата звернення: 10.10.2024)

17. Мокін В. Б., Коменчук О. В., “Сучасні інформаційні технології для розпізнавання дахів будівель на аерофотозйомці,” на LI Науково-технічної конференції факультету інтелектуальних інформаційних технологій та автоматизації Вінницького національного технічного університету, Вінниця, 31 травня – 1 червня 2022 р. 2022. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/16150/13565>. (Дата звернення: 10.10.2024).

18. Sydney Roofers, Roofing Company Sydney. Rooflines. Roof Types. 2024. URL: <https://www.rooflines.com.au/blog/6-most-common-australian-roof-types>. (Дата звернення: 10.10.2024).

19. Nexе-ua. Про монтаж черепиці. 2024. URL: <https://nexе-ua.com/ua/promontazh-cherepici/>. (Дата звернення: 10.10.2024).

20. Python. 2024. URL: <https://hyperhost.ua/info/uk/shho-take-python-i-navishho-vin-potriben>. (Дата звернення: 10.10.2024).

21. Kaggle. 2024. URL: <https://uk.wikipedia.org/wiki/Kaggle>. (Дата звернення: 10.09.2024).

22. Бібліотека NumPy. 2024. URL: <https://devzone.org.ua/post/chomu-vam-slid-vykorystovuvaty-numpy>. (Дата звернення: 10.10.2024).

23. Бібліотека Pillow. 2024. URL: <https://thetransmitted.com/ai/najkrashhibiblioteki-python-dlya-obrobki-zobrazhen/>. (Дата звернення: 10.10.2024)

24. Бібліотека Open CV. 2024. URL: <https://thetransmitted.com/ai/najkrashhibiblioteki-python-dlya-obrobki-zobrazhen/>. (Дата звернення: 10.10.2024).

25. Бібліотека PyTorch. 2024. URL: <https://uk.wikipedia.org/wiki/PyTorch>. (Дата звернення: 10.10.2024).

26. Бібліотека Albumentations. 2024. URL: <https://www.unite.ai/uk/albummentations-швидке-гнучке-збільшення-зображення-для-комп'ютерного-зору/>. (Дата звернення: 10.10.2024).

27. Бібліотека Matplotlib. 2024. URL:

<https://uk.wikipedia.org/wiki/Matplotlib>. (Дата звернення: 10.10.2024).

28. DE:TR: End-to-End Object Detection with Transformers. DETection TRansformer in GitHub. 2024. URL: <https://github.com/facebookresearch/detr>

29. Hans Thisanke, Chamli Deshan, Kavindu Chamith, Sachith Seneviratne, Rajith Vidanaarachchi, Damayanthi Herath, "Semantic Segmentation using Vision Transformers: A survey,". 2023. DOI: <https://doi.org/10.48550/arXiv.2305.03273>

30. Semantic Segmentation on ADE20K. Table and Graph in Paperswithcode. 2024. URL: <https://paperswithcode.com/sota/semantic-segmentation-on-ade20k>. (Дата звернення: 10.10.2024).

31. Object Detection on COCO test-dev. Table and Graph in Paperswithcode. 2024. URL: <https://paperswithcode.com/sota/object-detection-on-coco>. (Дата звернення: 10.10.2024).

32. Massachusetts buildings dataset – версія датасета – 01.10.2024 р. URL: <https://www.kaggle.com/datasets/balraj98/massachusetts-buildings-dataset>. (Дата звернення: 01.10.2024).

33. Аналіз міського середовища за допомогою K-Means. 2024. URL: <https://www.tandfonline.com/doi/abs/10.1080/10630732.2023.2270900>. (Дата звернення: 10.10.2024).

34. K-Means у ансамблі методів сегментації зображень. 2024. URL: <https://ieeexplore.ieee.org/abstract/document/10642907>. (Дата звернення: 10.10.2024).

35. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт, Вінниця : ВНТУ, 2021. 42 с.

36. Кавецький В. В., Козловський В.О., Причепка І. В. Економічне обґрунтування інноваційних рішень: практикум. Вінниця : ВНТУ, 2016. 113 с.

Додаток А
Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

«___» _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ ПОБУДОВИ ПЛАНУ БУДІВЕЛЬ
ПО АЕРОФОТОЗЙОМЦІ ЇХ ДАХІВ»

08-34.МКР.002.02.000.ТЗ

Керівник: к.т.н., доц. каф. САІТ

_____ Євгеній КРИЖАНОВСЬКИЙ

«___» _____ 2024 р.

Розробив: студент гр. ІСТ-23м

_____ Владислав БУДЯК

«___» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 2024 р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САІТ від «__» _____ 2024 р.

2. Джерела розробки:

– Qi Chen, Lei Wang, Yifan Wu, Guangming Wu, Zhiling Guo, Steven L. Waslander, TEMPORARY REMOVAL: Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 147. 2019. DOI: <https://doi.org/10.1016/j.isprsjprs.2018.11.011>.

– Мокін В. Б., Варчук І. В., Крижановський Є. М., Інформаційна технологія аналізу та оптимізації топологічної спостережуваності багатозв'язних геоінформаційних систем: монографія. 2019. URL: <https://press.vntu.edu.ua/index.php/vntu/catalog/book/522>.

– Зубик А. І., ГІС в урбаністиці та просторовому плануванні: навчально методичний посібник для аудиторної та самостійної роботи студентів з курсу “Використання ГІС в урбаністиці та просторовому плануванні”. 2021. URL: <https://geography.lnu.edu.ua/wp-content/uploads/2021/03/Zubyk-HIS-v-urbanistytsi-ta-prostorovomu-planuvanni-book.pdf>.

3. Мета і призначення роботи:

Метою роботи є підвищення точності та ефективності побудови планів будівель за даними аерофотозйомки.

4. Вихідні дані для проведення робіт:

- Kaggle Dataset „Aerial Imagery For Roof Segmentation” <https://www.kaggle.com/datasets/atilol/aerialimageryforroofsegmentation/data>;
- Kaggle Dataset „Massachusetts Buildings Dataset” <https://www.kaggle.com/datasets/balraj98/massachusetts-buildings-dataset>.

–

5. Методи дослідження:

- аналіз аерофотознімків;
- сегментація зображень;
- методи машинного навчання – використання підходів на основі Vision Transformers (ViT) та DETR.

6. Етапи роботи і терміни їх виконання:

- | | | | |
|---|-------|---|-------|
| a) Аналіз предметної області..... | _____ | – | _____ |
| b) Вибір оптимальних інформаційних технологій..... | _____ | – | _____ |
| c) Вибір мови програмування та середовища розробки..... | _____ | – | _____ |
| d) Розроблення інтелектуальної технології | _____ | – | _____ |
| e) Економічна частина | _____ | – | _____ |
| f) Оформлення пояснювальної записки..... | _____ | – | _____ |

7. Очікувані результати та порядок реалізації:

Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів, результатом якої є підвищення точності та ефективності побудови планів будівель за даними аерофотозйомки.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання магістерських кваліфікаційних робіт для студентів спеціальності 126 «Інформаційні системи та технології» (освітня програма «Інформаційні технології аналізу даних та зображень»)).

9. Порядок приймання роботи

Публічний захист «__» _____ 2024 р.

Початок розробки «__» _____ 2024 р.

Граничні терміни виконання МКР «__» _____ 2024 р.

Розробив студент групи ІСТ-23м _____ Владислав БУДЯК

Додаток Б

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інтелектуальна технологія побудови плану будівель по аерофотозйомці їх дахів»

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра САІТ

Керівник: Євгеній КРИЖАНОВСЬКИЙ, к.т.н., доц. каф. САІТ

Показники звіту подібності Turnitin

Оригінальність 89 %

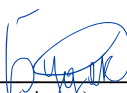
Загальна схожість 11 %

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи


(підпис)

Владислав БУДЯК

Опис прийнятого рішення

Робота допускається до захисту

Особа, відповідальна за перевірку


(підпис)

Сергій ЖУКОВ

Додаток В

Лістинг програми

```
!pip install segmentation_models_pytorch

import os
import sys

from tqdm import tqdm
import time
from datetime import timedelta

import random
import numpy as np

from PIL import Image
import cv2

import torch
import torch.nn as nn
import segmentation_models_pytorch as smp
from torch.utils.data import Dataset, DataLoader
import torch.optim as optim

import albumentations as A

from torchmetrics import JaccardIndex
import matplotlib.pyplot as plt

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

print(f"Device: {device}")

LEARNING_RATE = 0.001
BATCH_SIZE = 2
ENCODER_NAME = "mit_b2"
ENCODER_WEIGHTS = "imagenet"

def roof_processing(original_image, segmentation_mask, kernel_size=(3, 3), lower_limit=1, upper_limit=50):
    masked_image = cv.bitwise_and(original_image, segmentation_mask, mask=segmentation_mask)
    grayscale_image = cv2.cvtColor(masked_image, cv2.COLOR_BGR2GRAY)
    blurred_image = cv2.GaussianBlur(grayscale_image, kernel_size, 0)
    edged_image = cv2.Canny(blurred_image, lower_limit, upper_limit)
    return edged_image
```

```

def roof_ridge_detection(original_image, edged_image, hough_grid=1, threshold=15, min_line_length=5
0, max_line_gap=30):
    theta = np.pi / 180
    lines = cv2.HoughLinesP(edged_image, hough_grid, theta, threshold, np.array([]), min_line_lengt
h, max_line_gap)

    for line in lines:
        for x1, y1, x2, y2 in line:
            cv2.line(original_image, (x1, y1), (x2, y2), (0, 255, 0), 3)

    return original_image

def seed_everything(seed):
    random.seed(seed)
    os.environ["PYTHONHASHSEED"] = str(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False

seed_everything(42)

class DataRetriever(Dataset):
    def __init__(self, data_dir, data_list, transform=None):
        self.data_dir = data_dir
        self.data_list = data_list
        self.transform = transform

    def __len__(self):
        return len(self.data_list)

    def __getitem__(self, idx):
        if torch.is_tensor(idx):
            idx = idx.tolist()

        # image_path = f"{self.data_dir}/image/{self.data_list[idx]}"
        # image = np.array(Image.open(image_path).convert("RGB"))

        # print(self.data_dir)

        if "train" in self.data_dir:
            image_path = f"{self.data_dir}/image/{self.data_list[idx]}"
            mask_path = f"{self.data_dir}/label/{self.data_list[idx].split('.')[0]}_vis.tif"
        elif "val" in self.data_dir:
            image_path = f"{self.data_dir}/image/{self.data_list[idx]}.tif"
            mask_path = f"{self.data_dir}/label/{self.data_list[idx]}_vis.tif"

        image = np.array(Image.open(image_path).convert("RGB"))
        mask = np.array(Image.open(mask_path).convert("L"))
        mask = mask / 255.0

        if self.transform:
            augmentations = self.transform(image=image, mask=mask)
            image = torch.from_numpy(augmentations["image"]).permute(2, 0, 1)
            mask = torch.from_numpy(augmentations["mask"]).int().unsqueeze(0)

        return {"X": image, "y": mask}

```

```

DATA_DIR = "/kaggle/input/aerialimageryforroofsegmentation/"
TRAIN_DATA_DIR = "/kaggle/input/aerialimageryforroofsegmentation/train"
VALID_DATA_DIR = "/kaggle/input/aerialimageryforroofsegmentation/val"

TRAIN_TXT = f"{DATA_DIR}/train.txt"
VALID_TXT = f"{DATA_DIR}/val.txt"
TEST_TXT = "/kaggle/input/aerial-imagery-data-for-training/test.txt"

TRAIN_LIST = []
VALID_LIST = []
# TEST_LIST = []

# Open the file in read mode
with open(TRAIN_TXT, 'r') as file:
    # Read all lines from the file and store them in the list
    TRAIN_LIST = [line.strip() for line in file.readlines()]

# Open the file in read mode
with open(VALID_TXT, 'r') as file:
    # Read all lines from the file and store them in the list
    VALID_LIST = [line.strip() for line in file.readlines()]

# # Open the file in read mode
# with open(TEST_TXT, 'r') as file:
#     # Read all lines from the file and store them in the list
#     TEST_LIST = [line.strip() for line in file.readlines()]

# Define transforms for data augmentation
train_augs = A.Compose(
    [
        # A.Resize(512, 512),
        A.RandomCrop(always_apply=True, p=1.0, height=512, width=512),
        # A.GaussianBlur(),
        # A.RandomBrightnessContrast(),
        # A.HueSaturationValue(),
        # A.HorizontalFlip(),
        A.Normalize(
            mean=[0.0, 0.0, 0.0],
            std=[1.0, 1.0, 1.0],
            max_pixel_value=255.0,
            always_apply=True
        ),
    ],
)

valid_augs = A.Compose([
    # A.Resize(512, 512),
    A.RandomCrop(always_apply=True, p=1.0, height=512, width=512),
    A.Normalize(
        mean=[0.0, 0.0, 0.0],
        std=[1.0, 1.0, 1.0],
        max_pixel_value=255.0,
        always_apply=True
    ),
])

# Create the train dataset
#train_dataset = DataRetriever(data_dir=DATA_DIR, cases_list=TRAIN_LIST, transform=train_augs)
train_dataset = DataRetriever(data_dir=TRAIN_DATA_DIR, data_list=TRAIN_LIST, transform=train_augs)

# Create the valid dataset
#valid_dataset = DataRetriever(data_dir=DATA_DIR, cases_list=VALID_LIST, transform=valid_augs)
valid_dataset = DataRetriever(data_dir=VALID_DATA_DIR, data_list=VALID_LIST, transform=valid_augs)

```

```

train_dataset.__getitem__(0)

sample_idx = 8 # Change this to visualize a different sample
sample = train_dataset[sample_idx]
image, mask = sample["X"], sample["y"]

def visualize_sample(image, mask):
    plt.figure(figsize=(24, 24))

    # Plot the original image
    plt.subplot(1, 2, 1)
    plt.imshow(image.permute(1, 2, 0))
    plt.title("Original Image")
    plt.axis("off")

    # Plot the mask
    plt.subplot(1, 2, 2)
    plt.imshow(mask.permute(1, 2, 0), cmap='gray')
    plt.title("Mask")
    plt.axis("off")

    plt.show()

# Visualize the sample
visualize_sample(image, mask)

train_loader = DataLoader(train_dataset, batch_size=BATCH_SIZE, shuffle=True, drop_last=True)
valid_loader = DataLoader(valid_dataset, batch_size=BATCH_SIZE, shuffle=False, drop_last=False)

def build_model(num_classes):
    model = smp.MANet(classes=num_classes, encoder_name=ENCODER_NAME, encoder_weights=ENCODER_WEIGHTS,
    activation=None,)
    model.to(device)
    return model

model = build_model(num_classes=1)

num_classes = 1
num_epochs = 10

optimizer = optim.Adam(model.parameters(), lr=LEARNING_RATE)

criterion = smp.losses.JaccardLoss(mode="binary").to(device)
metric_fn = JaccardIndex(task="binary", threshold=0.5).to(device)

scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(optimizer, mode="min", factor=0.1, patience=
3, verbose=True)

```

```

def train_one_epoch(
    model,
    loader,
    criterion,
    metric_fn,
    optimizer,
    device,
    verbose,
) -> dict:

    # Set the model to training mode
    model.train()

    # Initialize lists to store losses and scores
    losses = []
    scores = []

    # Create a progress bar for training
    with tqdm(total=len(loader), desc="Training", file=sys.stdout, ncols=100, disable=not verbose)
    as progress:
        # Iterate over mini-batches in the DataLoader
        for mini_batch in loader:
            # Move data to the specified device (CPU or GPU)
            X_true = mini_batch["X"].to(device)
            y_true = mini_batch["y"].to(device)

            # Clear previous gradients (we want weights to accumulate, not gradients)
            optimizer.zero_grad()

            # Forward pass to compute predictions
            y_pred = model(X_true).sigmoid()

            # Compute the loss
            loss = criterion(y_pred, y_true)

            # Perform backward propagation to compute gradients
            loss.backward()

            # Append the current loss to the losses list
            losses.append(loss.item())

            # Compute and append scores
            score = metric_fn(torch.sigmoid(y_pred), y_true).item() # Compute the IoU/Dice score
            scores.append(score)

            # Update the progress bar with the current loss and score
            progress.set_postfix_str(f"Loss {losses[-1]:.4f}, Score {scores[-1]:.4f}")

            # Update the model's weights based on the optimizer's strategy
            optimizer.step()

            # Update the progress bar
            progress.update(1)

    # Check current LR
    current_lr = optimizer.param_groups[0]['lr']

    # Create a dictionary to store evaluation logs
    logs = {
        "losses": np.array(losses),
        "scores": np.array(scores),
        "lr": current_lr,
    }

    return logs

```

```

@torch.inference_mode()
def evaluate(
    model: nn.Module,
    loader: DataLoader,
    criterion: nn.Module,
    metric_fn,
    device: str,
    verbose: bool,
) -> dict:
    # Set the model to evaluation mode
    model.eval()

    # Initialize lists to store losses and scores
    losses = []
    scores = []

    # Create a progress bar for evaluation
    with tqdm(total=len(loader), desc="Evaluation", file=sys.stdout, ncols=100, disable=not verbose) as progress:
        # Iterate over mini-batches in the DataLoader
        for mini_batch in loader:
            # Move data to the specified device (CPU or GPU)
            X_true = mini_batch["X"].to(device)
            y_true = mini_batch["y"].to(device)

            # Forward pass to compute predictions
            y_pred = model(X_true) #.sigmoid()

            # Compute the loss
            loss = criterion(y_pred, y_true)

            # Append the current loss to the losses list
            losses.append(loss.item())

            # Compute and append scores
            score = metric_fn(torch.sigmoid(y_pred), y_true).item() # Compute the IoU/Dice score
            scores.append(score)

            # Update the progress bar with the current loss and score
            progress.set_postfix_str(f"Loss {losses[-1]:.4f}, Score {scores[-1]:.4f}")

            # Update the progress bar
            progress.update(1)

    # Create a dictionary to store evaluation logs
    logs = {
        "losses": np.array(losses),
        "scores": np.array(scores),
    }

    return logs

```

```

patience = 5
epochs_no_improve = 0

train_losses = []
train_scores = []

valid_losses = []
valid_scores = []

start_time = time.time()

best_ep = -1
best_val_loss = +np.inf
best_val_score = -np.inf
best_ep_ends_in = 'time'

for ep in range(num_epochs):
    print(f"\nEpoch {ep + 1:2d}/{(num_epochs:2d)}")

    train_logs = train_one_epoch(
        model,
        train_loader,
        criterion,
        metric_fn,
        optimizer,
        device,
        verbose=True
    )

    train_score = np.mean(train_logs["scores"])
    train_loss = np.mean(train_logs["losses"])

    train_scores.append(train_score)
    train_losses.append(train_loss)

    print("        train score:", train_scores[-1])
    print("        train loss:", train_losses[-1])
    print("        train lr:", train_logs["lr"])

    valid_logs = evaluate(
        model,
        valid_loader,
        criterion,
        metric_fn,
        device,
        verbose=True
    )

    valid_score = np.mean(valid_logs["scores"])
    valid_loss = np.mean(valid_logs["losses"])

    valid_scores.append(valid_score)
    valid_losses.append(valid_loss)

    print("        valid iou:", valid_scores[-1])
    print("        valid loss:", valid_losses[-1])

    # Early Stopping Check
    if valid_loss < best_val_loss:
        best_val_loss = valid_loss
        best_val_score = valid_score
        best_ep = ep + 1
        best_ep_ends_in = str(timedelta(seconds=time.time() - start_time))
        epochs_no_improve = 0

```

```

checkpoint = {
    "model": model,
    "model_state_dict": model.state_dict(),
    "optimizer_state_dict": optimizer.state_dict(),
    "epoch": ep,
    "min_valid_loss": best_val_loss,
    "best_valid_score": best_val_score,
    "train_losses": train_losses,
    "train_scores": train_scores,
    "valid_losses": valid_losses,
    "valid_scores": valid_scores,
}

print(f'Saving Best Model on epoch {ep + 1}')
torch.save(checkpoint, 'model_checkpoint.pth')

else:
    epochs_no_improve += 1

if epochs_no_improve >= patience:
    print(f'Early stopping triggered at epoch {ep + 1}')
    break

scheduler.step(valid_loss)

print(f'time for {ep+1} epochs:', best_ep_ends_in)
print(f'epoch no improve: {epochs_no_improve}')

print(f'\nBest epoch: {best_ep}')
print("Best validation loss:", best_val_loss)
print("Best validation score:", best_val_score)
print('time:', best_ep_ends_in)

```

ІЛЮСТРАТИВНА ЧАСТИНА

ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ ПОБУДОВИ ПЛАНУ БУДІВЕЛЬ ПО
АЕРОФОТОЗЙОМЦІ ЇХ ДАХІВ

Нормоконтроль: к.т.н., доцент



Сергій ЖУКОВ

« » 2024 р.



Рисунок Г.1 – Приклади з набору даних, де дахи займають незначну частину зображень



а



б



в

Рисунок Г.2 – Приклад з набору даних, де схожі види дахів розташовані в одній щільній забудові

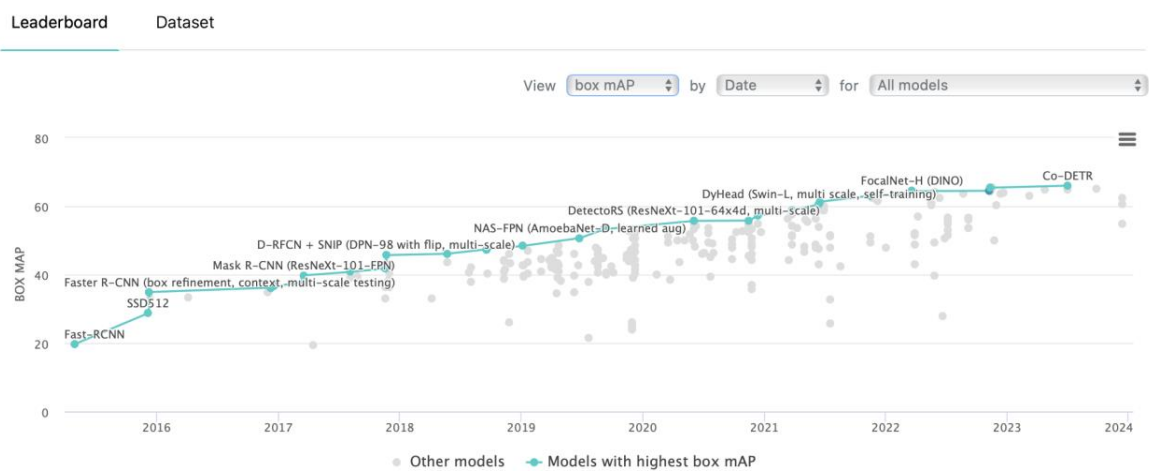


Рисунок Г.3 – Порівняння точності моделей для сегментації зображень

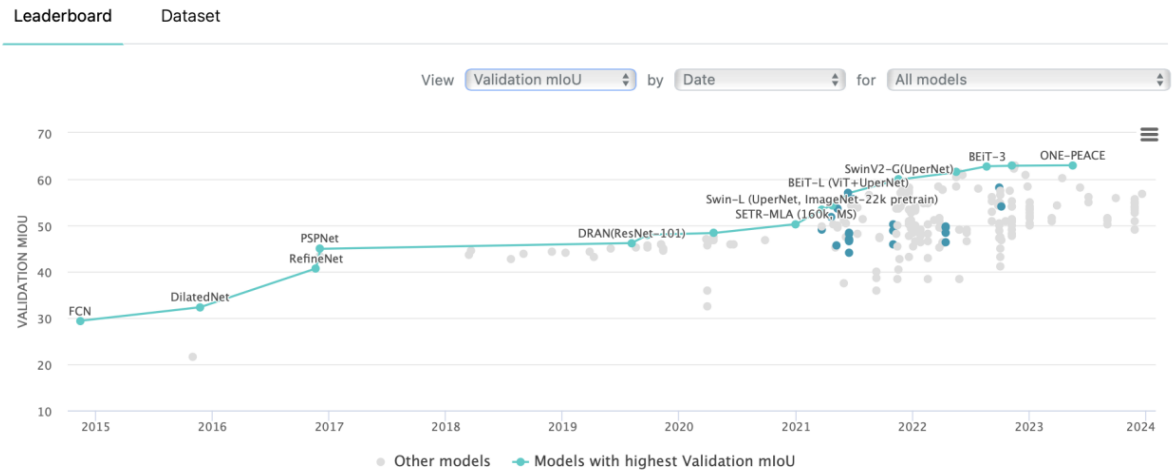


Рисунок Г.4 – Порівняння точності моделей для детектування об’єктів на зображеннях (синім виділено моделі класу “Transformers”)

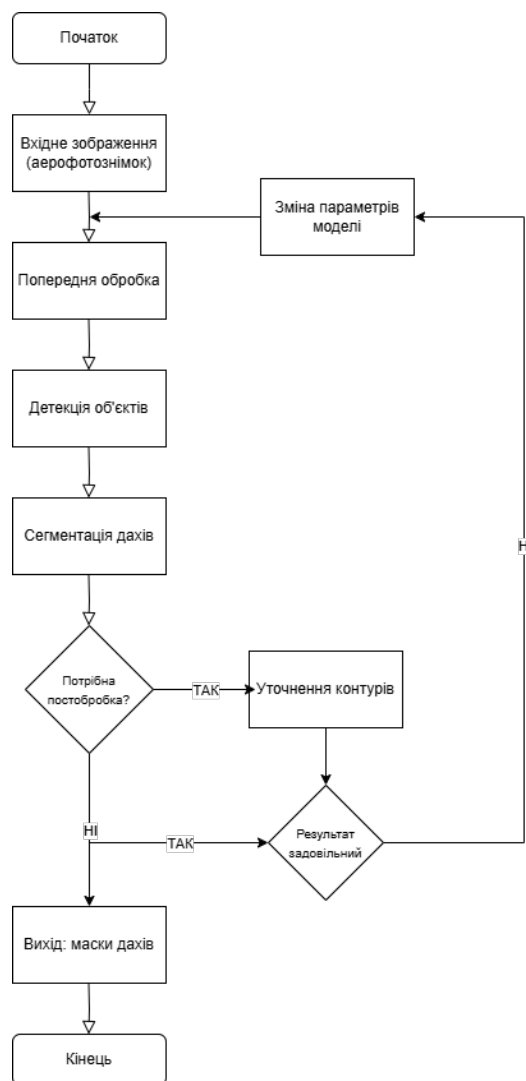


Рисунок Г.5 – Блок-схема алгоритму сегментації дахів

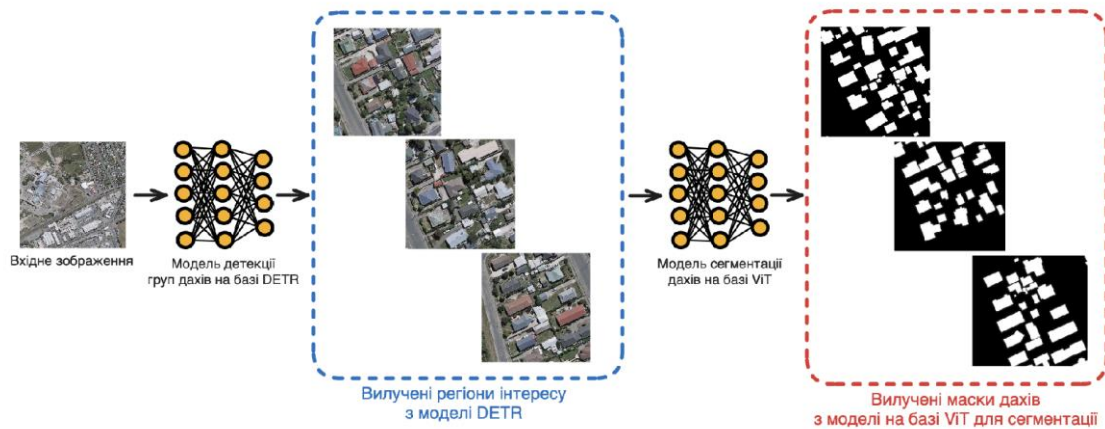


Рисунок Г.6 – Структурна схема архітектури інтелектуальної технології, що включає компоненти з використанням моделей DETR та ViTs

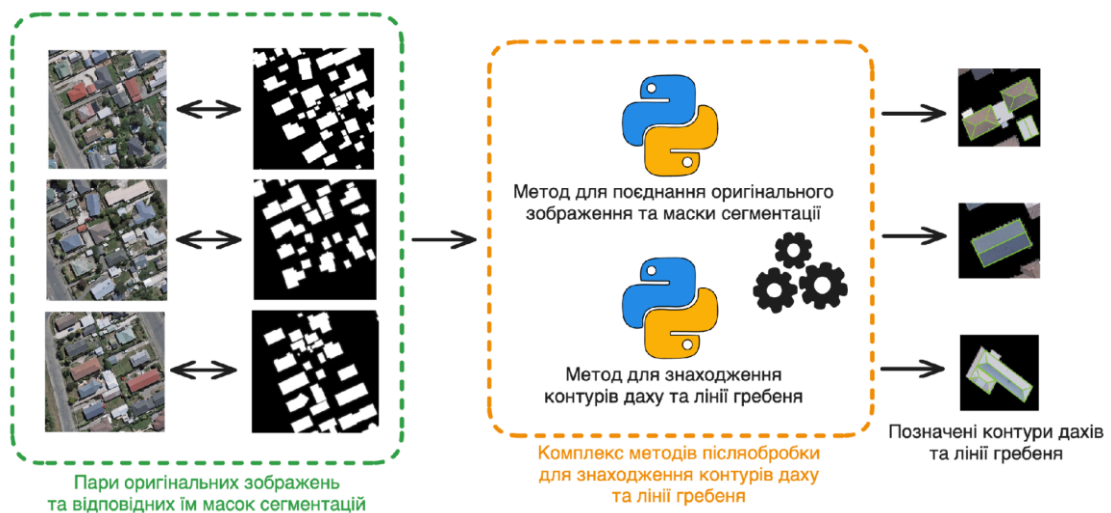


Рисунок Г.7 – Структурна схема архітектури інтелектуальної технології, що включає компоненти з використанням методів післяоброблення для знаходження контурів дахів та ліній гребеня