

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Центр _____ Післядипломної освіти
(повна назва)

Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

_____ Розробка системи класифікації картографічних зображень
_____ на основі згорткової нейронної мережі
(тема)

Виконав:
здобувач _____ другого _____ року навчання,
групи _____ ІТШпз-23-1

_____ Олеся Терлецька
(власне ім'я, прізвище)

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

Тип програми _____ освітньо-професійна
Освітня програма _____ Штучний інтелект
(повна назва освітньої програми)

Керівник _____ ст. викл. Філіп Бродецький
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____
(підпис)

_____ Олег ЗОЛОТУХІН
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Центр _____ Післядипломної освіти _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

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

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Терлецькій Олесі Олександрівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розробка системи класифікації картографічних зображень на основі згорткової нейронної мережі _____

затверджена наказом університету від 19 травня 2025 р. № 87Стз

2. Термін подання студентом роботи до екзаменаційної комісії 24 червня 2025 р.

3. Вихідні дані до роботи _____ Наукові публікації та академічні статті, Інтернет-джерела прикладного характеру, документація бібліотек мови програмування Python, підручники _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі і постановка задачі _____

2) Методи та технології розв'язання задачі _____

3) Практична реалізація _____

РЕФЕРАТ

Пояснювальна записка: 72 с., 18 рис., 1 табл., 5 дод., 37 джерел.

БІНАРНА КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КЛАСИФІКАЦІЯ КАРТОГРАФІЧНИХ МАТЕРІАЛІВ, ПЕРЕНЕСЕНЕ НАВЧАННЯ, ПОПЕРЕДНЬО НАВЧЕННІ МОДЕЛІ.

Кваліфікаційна робота присвячена дослідженню попередньо навчених моделей для згорткової нейронної мережі з метою розробки системи для класифікації картографічних матеріалів таких як мап.

Предметом дослідження є класифікація зображень за допомогою згорткової нейронної мережі, а об'єктом – попередньо навчені моделі у контексті перенесеного навчання для бінарної класифікації зображень з використанням згорткової нейронної мережі.

Метою роботи є дослідження попередньо навчених моделей для задачі бінарної класифікації картографічних зображень, налаштування моделей для найбільш ефективного результату. Серед методів дослідження, використані для освоєння роботи присутні емпіричні наукові методи (зокрема експеримент та порівняння), аналіз і синтез з ряду емпірико-теоретичних методів.

Головними завданнями роботи є дослідження академічного доробку у галузі комп'ютерного зору та класифікації зображень, проведення експериментів з трьома попередньо навченими моделями для вирішення двох завдань бінарної класифікації та побудова інформаційної системи типу вебзастосунка використовуючи дві моделі з відповідними налаштуваннями, що показали найкращий рівень класифікації.

ABSTRACT

Bachelor's thesis contains: 72 pp., 18 fig., 1 tabl., 5 ann., 37 references.

BINARY IMAGE CLASSIFICATION, CARTOGRAPHICAL MATERIALS' CLASSIFICATION, CONVOLUTIONAL NEURAL NETWORK, PRETRAINED MODELS, TRANSFER LEARNING.

The main focus of this thesis is the exploration of the pretrained model used for the convolutional neural networks in order to build the system for the binary classification of the cartographical materials such as maps.

The subject of the thesis is the classification of images using convolutional neural networks. The object of the thesis is pretrained models in the context of transfer learning for the binary classification using convolutional neural networks.

The goal of the thesis is the exploration of pretrained models for the task of binary cartographical materials' classification, in order to achieve the most highest results of classification. There were used mainly two type of scientific methods such as empirical (the experiment and the comparison methods) as well as empirical theoretical methods (analysis and synthesis).

The main tasks are research of the current scientific findings in the field of computer vision and image classification, conduction experiments using the three selected pre-trained models for two classification tasks and development of the system as a web application based on the two best models selected in the result of experiments.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Аналіз предметної галузі і постановка задачі	11
1.1 Комп'ютерний зір як напрямок дослідження у межах комп'ютерних наук	11
1.2 Згортова нейронна мережа: архітектура та призначення.....	14
1.3 Порівняння ЗНМ та візуального трансформера.....	16
1.4 Постановка задачі.....	18
2 Методи та технології розв'язання задачі	20
2.1 Перенесене навчання	20
2.2 Попередньо навчені моделі.....	22
2.3 Налаштування моделей	27
2.4 Оцінка роботи моделей.....	28
3 Практична реалізація	30
3.1 Опис вибірок завдання	30
3.2 Експериментальні дослідження.....	32
3.2.1 Розв'язання першого завдання	32
3.2.2 Розв'язання другого завдання.....	37
3.3 Розробка системи	40
3.3.1 Розробка UI/UX.....	41
3.3.2 Розробка бекенду	45
3.4 Перспективи розвитку системи	46
Висновки	47
Перелік джерел посилання	49
Додаток А Модель на основі попередньо навченої моделі MobileNet.....	54
Додаток Б Модель на основі попередньо навченої моделі EfficientNet.....	58
Додаток В Модель на основі попередньо навченої моделі ResNet.....	63
Додаток Г Таблиця результатів класифікації.....	69

Додаток Д Відомість кваліфікаційної роботи	72
--	----

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

АФ – активаційна функція;

БШП – багатошаровий перцептрон;

ВТ – візуальний трансформер;

ЗНМ – згорткова нейронна мережа;

ПНМ – попередньо навчені (натреновані) моделі;

ШІ – штучний інтелект;

ШНМ – штучна нейронна мережа.

ВСТУП

Предмет автоматизованої роботи з зображеннями є нагальним питанням у галузі розвитку комп'ютерних наук. Розвиток технологій, що дозволяють без людського ока збирати, анотувати, аналізувати, виокремлювати об'єкти на зображеннях та відео є вагомим та значущим для розвитку усіх сфер людського життя від розваги до безпеки життя.

На разі комп'ютерний зір активно застосовуються у сферах медицини, транспорту, військової справи та безпеки, охорони об'єктів чи людей, для розвитку соціальних мереж та телебачення. До прикладу, у медицині є попит на автоматизований аналіз рентгенівських, ультразвукових зображень для виявлення аномалій; у військовій справі за допомогою технологій комп'ютерного зору можливо вчасно виявити небажані чи небезпечні об'єкти на супутникових знімках; у сфері транспорту, нині активно тестується безпілотний автомобіль, а автоматизована анотація зображень у соціальних мережах, допоможе людям дізнатись інформацію навіть за умов обмеженого зв'язку з Інтернетом.

У контексті цієї роботи, дослідження методів класифікації картографічних зображень потенційно може бути застосовано для визначення реальних мап та штучно створених з метою дезінформації тощо; автоматизоване визначення часового періоду, в який певна карта була створена, а також визначення сучасних та історичних документів; автоматизована класифікація картографічних зображень для пошуку прихованих об'єктів або територій на них.

Предметом дослідження є попередньо навчені моделі у контексті перенесеного навчання для бінарної класифікації зображень з використанням згорткової нейронної мережі.

Об'єктом дослідження є попередньо навчені моделі у контексті перенесеного навчання для бінарної класифікації зображень з використанням згорткової нейронної мережі.

Метою роботи є дослідження попередньо навчених моделей для задачі бінарної класифікації картографічних зображень, налаштування моделей для найбільш ефективного результату.

Методи, використані для роботи включають емпіричні наукові методи такі як експеримент та порівняння (експерименти на основі налаштувань гіперпараметрів моделі для отримання кращих результатів класифікації, а порівняння застосовувалось для визначення зміни критерію чи індексу певного гіперпараметру моделі та вплив цієї зміни на результати), та емпірико-теоритичні такі як аналіз та синтез (полягає у оцінці зміни гіперпараметрів, вплив зміни на результат, узагальнення даних для наступних експериментів тощо). Завданнями роботи виступають:

- дослідити сучасний академічний доробок у галузі комп'ютерного зору та, власне, класифікації зображень як завдання у межах комп'ютерних наук;

- дослідити наявні попередньо навчені згорткові нейронні мережі для практичної частини дослідження; провести експерименти з моделями для результатів ефективною класифікації картографічних зображень у межах двох задач: класифікація зображень обрисів інфраструктури міст з американського та європейського материків без найменувань та класифікація зображень історичних та сучасних мап;

- створення системи класифікації картографічних зображень з користувацьким інтерфейсом на основі двох моделей, з найвищими показниками точності у підсумку виконаного попереднього завдання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Комп'ютерний зір як напрямок дослідження у межах комп'ютерних наук

Комп'ютерний зір як напрямок дослідження розвивається на основі глибокого навчання (що своєю чергою, розвивалась та надихалось біологією та відтворювалось за допомогою математики та фізики), та розвитком фотографії як такої, та відео. Узагальнено можна відмітити, що завдяки широкій доступності у сучасності засобів фіксацій зображень, а також широкого використання світлин та відео для фіксування інформації, у суспільства виникає потреба в автоматизованому аналізі задля запобігання рутинного аналізу таких документів. Тому історичний розвиток комп'ютерного зору базується на розвитку фотографії та відео, потребі людини в автоматизації монотонних завдань та розвитку глибокого навчання у контексті машинного навчання.

У 1970-х розвивається фундаментальна технологія розпізнавання контурів на основі визначення яскравості зображення та застосування технології для задачі розпізнавання людського обличчя на світлинах [1]. На сьогодні технологія розпізнавання контурів використовується для задач сегментації зображень, розпізнавання об'єктів, виокремлення ознак та зменшення зображення зі збереженнями пропорцій [2].

Останні дві декади ХХ століття відзначаються розвитком технологій для вирішення завдання розпізнавання об'єктів на світлинах та застосування технік машинного навчання для цих задач, до прикладу, використання Гаусівської змішаної моделі, що широко використовувалась для кластеризації та моделювання візуальних даних.

На початку ХХІ століття популярним стає використання моделей з методами опорних векторів для розпізнавання обличчя, і вже у 2017 році

компанія Apple виробляє на широкі маси телефон iPhone X, у якому застосована технологія FaceID, що дає власникам більшу захищеність.

Опісля, період, в якому активно розвивається архітектура згорткової нейронної мережі, яка на сьогодні масово використовується при роботі з зображеннями та створюється ряд застосунків для розв'язання задач зі фото. У період від 2010 року у комп'ютерному зорі вивчається використання попередньо навчених моделей (ПНМ) у контексті перенесеного навчання та використання моделей трансформерів, що були представлені у науковому доробку вчених компанії Google від 2017 року з першочерговим фокусом на обробку природної мови [3].

Ціллю комп'ютерного зору як наукової дисципліни є навчання штучного інтелекту з метою розпізнавання відео та зображень для їхнього розуміння, розпізнавання тощо; навчання мереж виокремлювати окремі елементи та ознаки з відео та світлин [4, с. 3].

Комп'ютерний зір як галузь науки про комп'ютер тісно пов'язаний з наукою про штучні нейронні мережі, та, безпосередньо, впливає на розвиток робототехніки. На розвиток комп'ютерного зору впливає як математика, фізика, так і біологія та природа людського інтелекту, а також розвиток фіксувальних інструментів – камер різного типу, що фіксують світлини та відео.

Типовими задачами комп'ютерного зору є розпізнавання об'єктів на статичних зображеннях чи знімках або відео; класифікація зображень; сегментація зображень – виокремлення певних об'єктів на світлинах; реставрація пошкоджених фотографій; кольоризація чорно-білих знімків; розпізнавання емоцій людини на основі міміки на світлинах та відео; розпізнавання та аналіз руху; розпізнавання жестів; розпізнавання силуету об'єкта на світлинці, та багато інших [5], [6].

Є чимало ситуацій, в яких використання задач комп'ютерного зору є корисним та практичним: медична сфера (класифікація зображення, виявлення патологій на зацифрованих рентгенівських знімках, і т.д.); у

сфері логістики та інвентаризації (для швидкого автоматизованого розпізнавання об'єктів); виробництво (автоматизоване розпізнавання дефекту виробленого товару на етапі виробництва) [7]; сфера безпеки та оборони (виявлення об'єктів ворогів на певній території на відео чи фото, розпізнавання відбитків пальців); агрокультура (для виявлення дозрілого врожаю без фізичної перевірки на полях; визначення якості поливу чи ґрунту на основі зображень); доповнена реальність [8]; та багато інших застосувань, що вирішують багато проблем та ускладнень у робочому та особистому житті людини. Популярним напрямком розробки з застосуванням комп'ютерного зору наразі також є автоводій для авто.

Зокрема, задачі комп'ютерного зору реалізуються у таких відомих застосунках: Google Translate, власне, функція, яка дозволяє користувачу перекласти текст на зображенні, яке застосунок отримує у результаті роботи камери мобільного пристрою [9]. Apple Vision приклад практичного застосування доповненої реальності.

Можливо також окреслити сучасні виклики у роботі з комп'ютерним зором, а саме:

- просторова складність: з зображеннями, що є пласким відображенням об'ємних об'єктів [10], [11];
- різноманітність візуальної репрезентації: перспектива, освітлення та тінь;
- цілісність зображень може бути порушена стисненням чи сенсорним шумом;
- оклюзія об'єктів: коли один суб'єкт чи об'єкт у русі перекриває частину об'єкта спостереження;
- варіативність поміж значеннями у певній категорії може становити більший ризик працюючи з зображеннями, адже зі ними важче визначити межі однієї категорії від іншої, подекуди й людському оку.

1.2 Згорткова нейронна мережа: архітектура та призначення

Згорткова нейронна мережа (ЗНМ; convolutional neural network; CNN) – тип нейронної мережі, що у своїй архітектурі має декілька прихованих шарів. Цей тип НМ скерований на опрацювання даних у вигляді сітки (grid), що дозволяє ефективно аналізувати 2D зображення. Саме ця характеристика відрізняє цей тип НМ від класичної ШНМ, що приймає у якості вхідних даних саме одно просторовий масив [4, с. 266].

ЗНМ це така НМ, у якій застосовується згортання замість звичайного матричного множення у принаймні одному шарі [12, с. 330]. Згортка, як механізм, надає більшу вагу новішим даним (пікселям у контексті роботи зі зображеннями).

Багатошаровий перцептрон також має можливість обробити зображення, проте тільки в тому випадку, якщо вхідна матриця (зображення) буде перетворена у простий числовий вектор без просторової структури. Тоді як ЗНМ розуміє, що сусідні пікселі зображення мають сильніший зв'язок, ніж пікселі, що знаходяться на більшій відстані один від одного. Також різниця між ЗНМ та багатошаровим перцептроном полягає у розташуванні прихованих шарів: ЗНМ розташовує нейрони у трьох вимірах: в ширину, в довжину та у висоту [13, с. 33].

У ЗНМ виокремленні кілька шарів, що мають різне призначення: вхідний шар (в якому перебуває певне зображення, можливо кілька зображень), згортковий шар (головна ціль якого – вилучення ознак зображення з вхідного шару; тут також відбувається найбільша частина обрахунків), шар об'єднання (метою цього шару є зменшення обчислень за допомогою зменшення зв'язків між шарами; на цьому етапі обробки підсумовуються об'єкти, що є на зображенні).

Повністю об'єднаний шар, зазвичай, слідує за кількома згортковими та шарами об'єднання. Таке розміщення різних типів шарів пояснюється необхідністю це вилучення класів різних зображень на різних абстрактних

шарах. Останній, повністю об'єднаний шар, сплющується, й у результаті виходить 1D вектор [4, с. 270]. Архітектура ЗНМ представлена на рисунку 1.1.

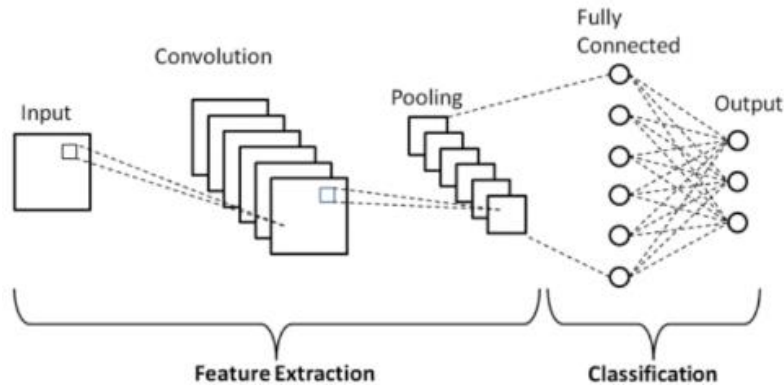


Рисунок 1.1 – Архітектура згорткової нейронної мережі

Серед переваг використання ЗНМ є наступні:

- вилучення ознак з вихідних даних автоматично у процесі навчання;
- використання просторових особливостей зображення на відміну від багатозарового персептрона, де простір як елемент зникає;
- спільне використання параметрів.

Недоліками системи є складність побудови та підтримки, а також порівняно низька швидкість, що зумовлена кількістю прихованих шарів та вхідних даних.

Практичне використання ЗНМ можливо для таких цілей як класифікація зображень, сегментація зображень, визначення об'єктів на зображенні, вилучення ознаки зображення тощо. Зокрема, популярна соціальна мережа Instagram використовує ЗНМ для опису об'єктів на зображеннях (що може відобразитись для користувача як список термінів, що характеризують зображення під час низького сигналу Інтернету на пристрої) [14]. Також прикладом практичного застосування ЗНМ є застосунок Google Lens, за допомогою якою користувачі мають змогу ідентифікувати об'єкт через камеру власного мобільного

пристрою (наприклад, що це є за рослина). Так, мережа аналізує завантажене зображення та класифікує зображений об'єкт, та видає користувачу клас об'єкта – вихідні дані [15].

1.3 Порівняння ЗНМ та візуального трансформера

Як було згадано вище, від 2017 року активно розвиваються моделі візуального трансформера (ViT). На сьогодні визначають ряд плюсів використання саме цих моделей для різних завдань, що пов'язані з обробкою та аналізом зображень. Серед моделей ВТ можна перелічити наступні: DETR, iGPT, ViT, які були випущені у 2020 році. Зазначається, що моделі ВТ вимагають меншого обчислюваного ресурсу на етапі попереднього навчання, та має слабше індуктивне зміщення, що потребує більшої регуляції або доповнення даних при роботі з невеликими вибірками.

Архітектура моделей ВТ містить механізм самоуваги (англ. «self-attention mechanism»). Механізм уваги полягає в імітуванні когнітивної уваги, та обчислює диверсифіковані або «м'які» ваги для вхідних даних (у контексті світлин – для кожного вхідного пікселя) на основі зв'язків конкретного пікселя з іншими, тоді як шар прямого поширення застосовує нелінійне перетворення до виходу шару самоуваги [16]. Трансформерні моделі таким чином фокусуються на просторових залежностей даних, та використовують послідовну обробку даних.

Тоді як ЗНМ використовують механізм згортки, де більша вага надається новішим даним (пікселям), а оброблюються прийняті масиви даних [17]. Архітектура візуального трансформера представлена на рисунку 1.2.

У порівнянні з ЗНМ, навчання ВТ з нуля вимагає більших ресурсів, що є результатом відсутності механізму згортки в архітектурі та відповідно меншої кількості специфічних індуктивних упереджень.

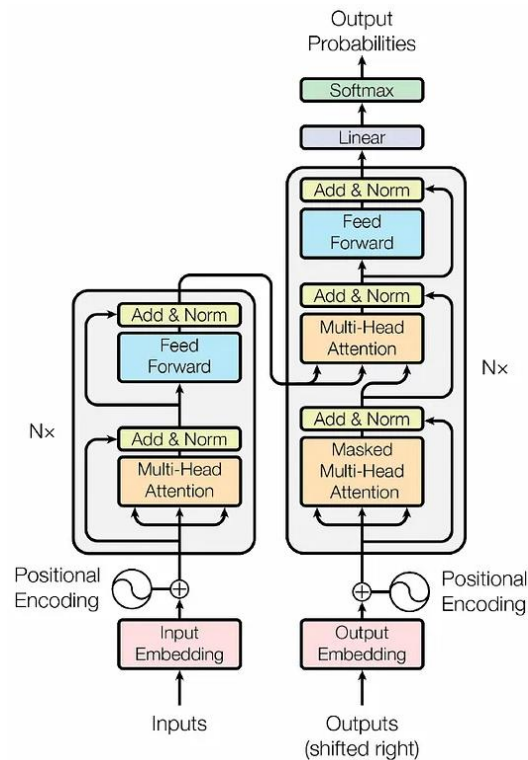


Рисунок 1.2 – Архітектура візуального трансформера [18]

Саме у ситуаціях, де модель має бути навчена з нуля, ефективність ЗНМ може бути вищою за ВТ у контексті ресурсів. ВТ також варто розглядати для використання у контексті моделі у випадку оперування великими об'ємами вибірок, а також у тому випадку, коли очікується зсув даних, що може відбуватись коли навчальні дані та цільові дані є відмінними, тож результати класифікації можуть бути нижчими через цей ефект. Власне, коли важлива високий результат класифікації, з обмеженим часовим ресурсом варто вдаватись до ЗНМ [19].

На рисунку 1.3 представлено порівняння між ЗНМ та ВТ у контексті кореляції між точністю та розміром моделі.

Чи є трансформерні чи згорткові нейронні мережі більш ефективними для розв'язання задач комп'ютерного зору залежить від факторів ресурсів навчання, використання техніки перенесеного навчання, розмірів цільових даних (для класифікації, сегментації, визначення об'єктів тощо).

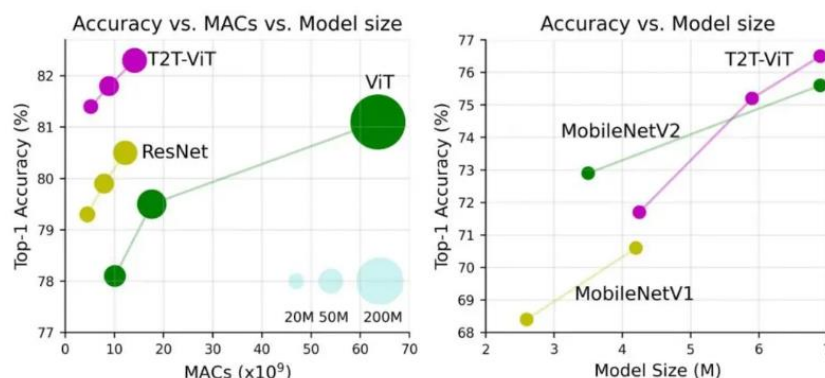


Рисунок 1.3 – Порівняння моделей ЗНМ та моделей VT

Загалом, розв'язання реальних проблем та задач з обмеженими цільовими даними популярним рішенням є початок роботи з ЗНМ з перенесеним навчанням; проте у випадку різної різниці між цільовими даними та даними, на яких модель була навчена, рекомендується застосувати саме трансформерні моделі та згодом порівняти з результатами ЗНМ [19].

1.4 Постановка задачі

Для виконання роботи, а саме експериментальної її частини були визначені три різних ПНМ, на основі яких відбувалась побудова системи класифікації: MobileNet, EfficientNet та ResNet. Перевагами для застосування саме ПНМ MobileNet була визначена легкість моделі у зв'язку ваги та швидке обчислення у контексті випробування на вибірках великого об'єму. Власне, легкість моделі та складність архітектури дозволяє такій моделі працювати з малими вибірками без остраху перенавчання, що може статись у таких випадках з моделями зі складнішою архітектурою.

ПНМ ResNet найкраще використовувати з вибірками великого розміру, адже така модель може вивчати складніші патерни, а великою перевагою її є пропуск залишкового з'єднання, що запобігає проблемі зникання градієнта.

EfficientNet може краще працювати з меншими вибірками (особливо порівнянні з ResNet) у площині перенавчання, проте врешті краще функціонують на більших вибірках.

Задачами кваліфікаційної роботи у вступі визначені три завдання, вирішення яких покликані створити систему класифікації. Завданнями теоретичної частини виступають дослідження перенесеного навчання, попередньо навчених моделей для високих результатів класифікації. Частина роботи про методи та технології розв'язання задачі також досліджуватиме особливості налаштування моделі, включно з гіперпараметрами, використання технік як доповнення даних та замороження шарів. Частина роботи, що сфокусована на експериментах з ПНМ моделями покликана перевірити тези (такі як, конкретна модель працює ліпше з великими вибірками або з меншими) та дослідити влучні гіперпараметри для конкретної моделі та їх навчання по епохах.

Основною метою цієї роботи є розробити систему для бінарної класифікації картографічних зображень на основі згорткової нейронної мережі та з використанням методів перенесеного навчання.

Завданнями до виконання є дослідити предметну область, а саме доступні застосунки та інструменти для класифікації зображень, дослідження та експерименти з різними ПНМ для вирішення двох завдань класифікації та використання вибраних моделей з найвищими показниками для побудови системи.

2 МЕТОДИ ТА ТЕХНОЛОГІЇ РОЗВ'ЯЗАННЯ ЗАДАЧІ

2.1 Перенесене навчання

Перенесене навчання це техніка машинного навчання, при якій модель самостійно не тренується, а використовуються готові моделі, які були попередньо навчені на великих даних.

Опісля використання перенесеного навчання та оцінки результатів роботи базової версії ПНМ, модель може пройти через крок тонкого налаштування (англ. «fine-tuning»), і хоча після тонкого налаштування, точність роботи моделі є вищою, цей крок не є обов'язковим [20].

Існують декілька типів перенесеного навчання:

– індуктивне перенесене навчання: у випадках, коли задача вихідної моделі (з джерела) та цільова задача відрізняються, проте простори ознак та міток є подібними або однаковими (у контексті зображення простори ознак – кількість пікселів у зображеннях, ознаки текстур, ознаки форми, і т.і., а простори міток – класи, анотації значень у вибірці);

– трансдуктивне перенесене навчання: тип навчання, коли цільова та вихідна задачі відрізняються, і так само простори та ознаки міток також є відмінними, а ПНМ застосовується для покращення навчання цільової моделі;

– некероване перенесене навчання: тип навчання, при якому значення з вибірки джерела не мають значень класів чи анотацій, а цільові дані мають цю інформацію, і знання використовуються для вилучення ознак, які корисні для цільової задачі;

– самонавчання: коли завдання без анотованих даних першочергово навчається для іншого завдання на основі даних, що мають такий самий або подібний розподіл, а знання, отримані з першого навчання, використовуються для виконання другого;

– багатозадачне навчання: в цьому випадку, одна модель навчається для виконання двох або більше задач одночасно, а спільні знання використовуються для ефективного розв'язання усіх задач [21]. На рисунку 2.1 представлена візуалізація перенесеного навчання.

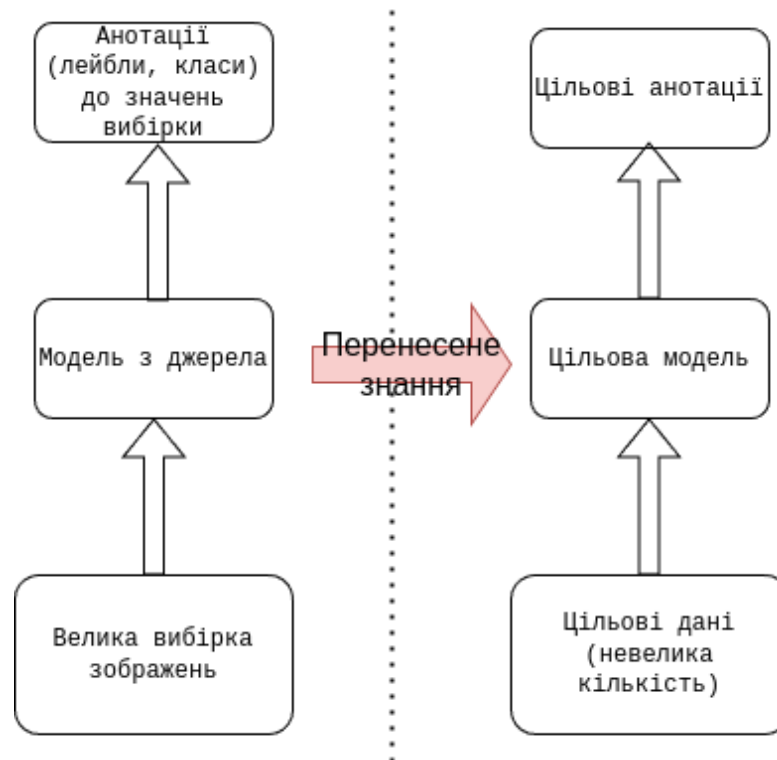


Рисунок 2.1 – Візуалізація перенесеного навчання

Багатозадачне навчання та самонавчання, подекуди виокремлюються як просунуті методи для досягнення кращих результатів моделі для цільового завдання. Також до переліку просунутих технік долучають: адаптація предметної області (саме коли вихідне завдання та дані не є подібними до цільових; така адаптація може допомогти досягти кращих результатів, аніж при використанні традиційного тонкого налаштування); навчання з малою кількістю прикладів (англ. «few shot learning»), у якому передбачається навчання з невеликою кількістю анотованих даних); навчання з нуля (англ. «zero-shot learning»), де обробляються немарковані

дані, а навчання відбувається за допомогою допоміжної інформації, до прикладу, семантичні зв'язки тощо [22].

2.2 Попередньо навчені моделі

Попередньо навчені моделі – моделі машинного навчання, які попередньо були натреновані (навчені) на великих об'ємах даних з метою тонкого налаштування мережі; використання таких моделей для подальшої побудови моделі для конкретного завдання, є зручним, адже ПНМ надають первинні значення ваг [23].

Використання ПНМ є особливо корисним у завданнях, які є досить комплексними, проте при цьому не мають доступу до великих масивів даних та інших ресурсів. Такі моделі можуть бути використані для розв'язання задач комп'ютерного зору: класифікація зображень, сегментація зображень, виявлення об'єктів на світлині, а також завдання з комплексами обробки природної мови (NLP). І саме такі моделі широко використовуються при застосуванні перенесеного навчання.

У контексті комп'ютерного зору, ПНМ працюють наступним чином:

- збирається широка колекція зображень на різноманітні теми (класи);
- модель тренується на вибірці, напрацьовуючи ваги та зменшуючи помилки для передбачення;
- модель навчається виокремлювати ознаки на зображеннях різних класів;
- модель може бути пристосована відповідно до завдання чи даних, які містяться у цільовій вибірці;
- етап тонкого налаштування;
- вилучення ознак зі значень цільової вибірки та результат [21].

Багато ПНМ, що створюються маючи на меті роботу зі світлинами або відео, навчаються на вибірці ImageNet. Ця вибірка містить близько 14 млн

анотованих зображень [24] та має зображення з тисячі категорій. Серед таких моделей можна згадати: ResNet, VGG, Inception, and MobileNet. U-Net моделі також використовуються для роботи з комп'ютерним зором, а саме для проблеми сегментації зображення. YoLo моделі використовуються для задач визначення об'єкта на зображеннях. CycleGAN моделі використовують для перетворення зображень на інший стиль, підлаштовуючи зображення під іншу предметну область [25].

Для завдань пов'язаних з обробкою та аналізом природної мови використовують ПНМ BERT (Bidirectional Encoder Representations from Transformers) для завдань типу класифікація тексту, відповідь на питання тощо. Аналогічно, використовуються моделі з ряду GPT (Generative Pretrained Transformer).

Наразі існує ряд ПНМ, які пристосовані до класифікації та інших завдань пов'язаних з зображеннями. До прикладу це AlexNet, VGG16, Inception, Xception, DenseNet, NASNet, SqueezeNet.

Для розв'язання двох завдань класифікації були використані три моделі: MobileNet, EfficientNet та ResNet. Модель MobileNet була використана через свою легкість та швидкість обчислення, тоді як модель ResNet була використана через свою здатність навчання складних патернів (саме для першого завдання, де виднілись невідповідна випадкова інфраструктура особливо європейських міст розглядається як хороший приклад), а EfficientNet через оптимальні результати.

MobileNet – категорія ПНМ, що включає три версії (MobileNetV1, MobileNetV2 та MobileNetV30. MobileNetV1 – модель натренована на вибірці ImageNet (ILSVRC). Така модель містить 1–4 млн параметрів. Архітектура моделі базується на інвертованій залишковій структурі, де вхід та вихід залишкового блоку є тонкими вузькими шарами [26].

Різниця між першої версією моделі та другою полягає у змінах в архітектурі: були введені інвертовані залишкові блоки та лінійні вузькі місця, а активаційна функція змінена з ReLu на ReLu6. У порівнянні з

першою, друга версія досягає точніших результатів та є швидшою. Щодо третьої версії, то є наявні два варіанти: менша та більша модель. У порівнянні з другою версією, менший варіант був точніший на 6,6% на вибірці ImageNet з подібним значенням затримки, коли більша версія на 3,2% та зменшує затримку на 20% [27]. На рисунку 2.2 представлена інформація щодо архітектури ПНМ MobileNet, а саме її третьої важкої версії.

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Рисунок 2.2 – Архітектура MobileNetV3 large

EfficientNet – ПНМ, що була розроблена GoogleAI у 2019 році. Архітектура цієї моделі рівномірно масштабує виміри даних (глибина, ширина, роздільна здатність) за допомогою техніки складного коефіцієнта, замість випадкового збільшення параметрів зображень [28], [29].

У роботі Т. Ахмеда від 2020 р. презентована архітектура базової версії ПНМ EfficientNet (рисунок 2.3), а також використані різні версії моделі (B0–

В5) для задачі класифікації хмарних структур на супутникових світлинах. У цій роботі також порівнюються результати роботи моделі з базовими налаштуваннями та підлаштованими, різниця між точностями різних версій моделей та їхніми налаштуваннями коливається між 13% та 17% [30].

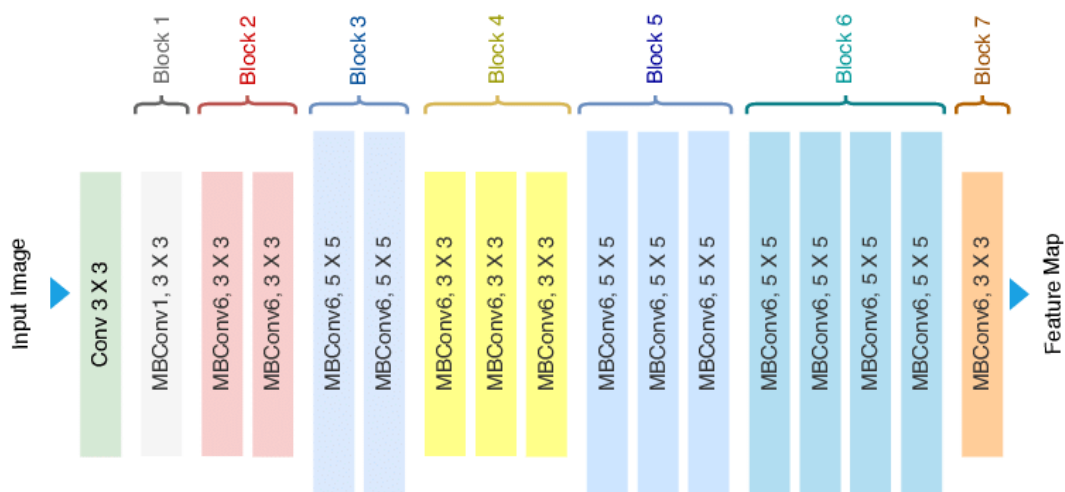


Рисунок 2.3 – Архітектура попередньо навченої моделі EfficientNet

ResNet це ПНМ запропонована в академічній статті Хе Каймінг et. al., від 2015 року «Deep Residual Learning for Image Recognition» [31], [32]. Основна ідея цієї моделі є використання залишкових блоків, а саме: кожний такий блок містить два чи три згорткових шари та пропускає з'єднання, яке оминає ці шари й додає вхідні дані x до вихідних (рисунок 2.4).

Плюсами використання цієї ПНМ є вища точність моделі у порівнянні з моделями, що містять у своїй архітектурі повністю об'єднані шари (наприклад, багатошаровий перцептрон); ResNet популярний вибір для перенесеного навчання, особливо у завданнях пов'язаних з обробкою природної мови; така модель має швидшу збіжність, яка забезпечує швидке навчання та менші витрат ресурсів. Проте, використання такої моделі з невеликим набором даних може призвести до перенавчання.

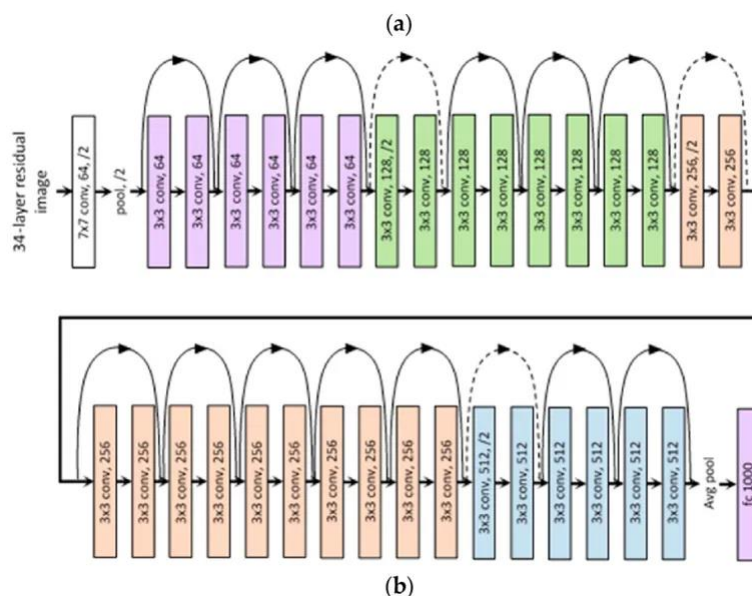


Рисунок 2.4 – Архітектура моделі ResNet з 34 шарами

Через те, що відбувається пропускання зв'язку збільшується складність моделі, та виконання завдання вимагає більшої оперативної пам'яті, та, загалом, використання ресурсів [33].

Всі вищезгадані ПНМ навчені на узагальнених зображеннях, та спеціально не містять мап у першочергових вибірках.

Проте на сьогодні існують спеціалізовані комерційні моделі, що були навчені на картографічних даних та згруповані за наступними типами завдань:

- виділення та виявлення особливостей зображення (такі моделі на даних автомобілів ЗША та Нової Зеландії, виділення дерев на мапах, визначення басейнів на знімках зі супутника тощо);

- класифікація пікселів (який піксель належить якому об'єкту; моделі, до прикладу, що визначають тип покриття поверхні, класифікація пішохідних зон, класифікація стоянок і т.д.);

- класифікація хмари точок (класифікація будівельних точок, точок дерев);

– редагування зображення (розмиття облич задля безпеки, розмиття номерів автівок);

– відстеження об'єктів у русі (відстеження руху людей з зображень дрона, світлини знімків будівель) [34].

Такі моделі доступні не тільки на комерційній основі, але й деякі є у безплатному доступі, розроблені компанією Esri.

2.3 Налаштування моделей

Експерименти з різними моделями полягали на зміні параметрів моделей. Були виокремлені такі категорії параметрів: архітектура моделі, конфігурація навчання, параметри пов'язані з даними, гіперпараметри, налаштування оптимізації. І відповідно, були занотовані результати навчання та процес навчання.

Були виокремлені наступні параметри архітектури моделі: назва використаної ПНМ, використання замороження шарів моделі. Замороження шарів у моделі при використанні перенесеного навчання може запобігати виникненню перенавчання (англ. «overfitting»), особливо якщо використовуються доволі мала цільова вибірка.

Оскільки, основною метою використання перенесеного навчання є використання вироблених знань з вихідної задачі та вибірки, то при умові не використання замороження шарів, модель може перезаписати попередньо вироблені знання [35]. Проте, інколи зазначається, що ігнорування заморожених шарів може мати позитивний ефект у тому випадку, коли цільові дані є досить відмінними від вихідних [36].

У категорії параметрів конфігурації навчання були виокремлені такі параметри як пропорція розмірів та об'ємів значення для навчання, валідації та випробування. А також розмір пакета (англ. «batch size») використовується для визначення кількості записів в одній ітерації навчання моделі.

Для експериментів з даними були використані такі параметри як розмір зображень та доповнення даних (англ. «data augmentation») – процес штучного генерування даних для навчання моделі. Перевагами доповнення даних є підвищення продуктивності моделі коштом збагачення набору даних та ознайомлення моделі з різними особливостями цільових даних; зменшення залежності від даних, що є особливо корисним коли цільові дані не є великими; зменшення перенавчання моделі, що також є поширеним явищем з маленькими вибірками [37].

Були також досліджені гіперпараметри моделі, а саме: кількість нейронів у повністю об'єднаному шарі (щільний шар), функція активації нейронів, кількість епох, рання зупинка. Рання зупинка це техніка, що допомагає запобігати перенавчанню шляхом спостереження результатів валідації; кількість нейронів у повністю об'єднаному шарі може також впливати на перенавчання (забагато нейронів) та недостатнього навчання моделі; активаційні функції дозволяють нейроном вивчати складніші зв'язки та представляє нелінійність, а кількість епох визначає з якою кількістю даних з вибірки модель зможе ознайомитись.

У категорії оптимізаційних налаштувань виокремлюємо наступні категорії: оптимізаційну функцію, що визначає як модель оновить ваги під час навчання на основі розрахованих втрат; швидкість навчання, що визначає розмір кроку під час градієнтного спуску; та функцію втрати.

2.4 Оцінка роботи моделей

Для оцінки роботи моделі були виведена наступна інформація: класифікаційний звіт та записи по епосі (прогрес навчання по кожній епосі моделі). За допомогою класифікаційного звіту визначалось чи збалансовані дані класу для класифікації (по значенню підтримки (англ. «support») для кожного класу), та який результат передбачення для кожного класу. Зокрема, бралися до уваги такі параметри з класифікаційного звіту як

точність по класу (англ. «precision»), що визначає наскільки точними є позитивні передбачення; повнота (англ. «recall»), що допомагає визначити скільки позитивних значень було визначено з усіх наявних у вибірці та F1-міра (англ. «F1-score»), що є сукупним значенням точності по класу та повнотою, покликана представити єдине значення результату класифікації та нівелює велику різницю між цими значеннями (якщо наявні).

З цього ж звіту і виокремлювалась загальна точність моделі для швидкої оцінки роботи моделі.

Також зберігалась інформація щодо прогресу навчання по епосі, що дозволяло глибоко проаналізувати точність роботи моделі на певній епосі, поступова втрата та валідаційні показники. Аналіз цієї інформації дозволяв визначення перенавчання та відредагувати загальну кількість епох, якщо модель навчалась швидко або повільно.

Додатково були генеровані графіки точності та втрати моделі у процесі всього навчання. Подекуди вручну підраховувалась матриця невідповідностей на основі попередньо згенерованого класифікаційного звіту.

Результати роботи моделей були збережені у трьох різних Google Sheets файлах, а інформація щодо кожного експерименту зберігалась на окремому листі для доступу у будь-який момент.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Опис вибірок завдання

Для порівняння ПНМ у рамках бінарної класифікації було запропоновано два завдання: фаз (рисунок 1.3). Ці системи можна розділити на такі компоненти:

- бінарна класифікація обрисів міст з американських материків та європейської частини Євразії;
- бінарна класифікація історичних та сучасних мап.

Перше завдання характеризується малою вибіркою. Підсумовуючи, обидва класи налічують 117 екземплярів (57 американські міста та 60 європейські). Всі зображення у цій вибірці були зібрані вручну за допомогою онлайн інструменту який дозволяє вимкнути всі написи на мапах та знайти необхідні міста використовуючи пошукове поле. Стратегія збору зображень полягала у пошуку міст від найбільш заселених до менш заселених.

Друге завдання полягало у використанні вибірки середнього розміру. Таким чином, вибірка включає першочергово 12109 зображень (5808 сучасні мапи та 6301 історичні). Поділ на історичні та сучасні мапи загалом є умовним. Збір даних був автоматизованим, а джерелом зображень стала Вікіпедія та дві її категорії – Історичні та Сучасні мапи країн. Після того, як зображення для обох класів були зібрані, відбувся перегляд теки сучасні мапи на вміст історичних мап. Під «сучасними мапами» мається на увазі зображення, для створення яких були використані, власне, комп'ютерні програми або мапи, які можуть бути доступні при використанні пошуку в Інтернеті. Детальніша інформація щодо джерел та кількості екземплярів зображення зібраних чотирьох класів для обох вибірок представлені у таблиці 3.1.

Таблиця 3.1 – Опис вибірок

Завдання	Класи	Кількість зображень	Розміри	Методи збору	Джерело
1-ше завдання	Америка	57	1244x1116 px	ручний	snazzymaps
	Європа	60	1244x1116 px	ручний	snazzymaps
2-ге завдання	Сучасні мапи	5772	різні	автоматичні	Вікіпедія
	Історичні мапи	4810	різні	автоматичні	Вікіпедія

Загалом, історичні мапи з більшою ймовірністю були заскановані з паперових матеріалів, які з часом жовтіли, та рідше використовують яскраві кольори, на відміну від сучасних зображень мап, які часто використовують багато кольорів для кодування значень та подекуди зберігаються у форматі csv для збереження прозорого тла. До прикладу, відібрані мапи на рисунку 3.1, де є помітна різниця між історичними та сучасними мапами.

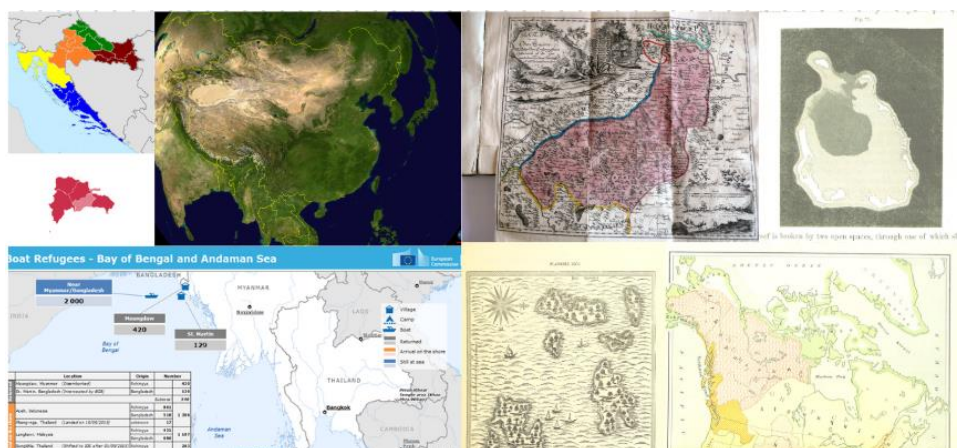


Рисунок 3.1 – Приклади мап сучасних (зліва) та історичних (справа)

Також, часто досить старі карти супроводжувались різноманітними зображенням та статтями.

3.2 Експериментальні дослідження

У цьому підрозділі описані два завдання, які покликані бути розв'язаними за допомогою налаштованих ПНМ у контексті застосування методу перенесеного навчання.

3.2.1 Розв'язання першого завдання

Для вирішення першого завдання – класифікація міст з європейського та американських континентів було відтворено 10 експериментів з різними параметрами на основі ПНМ MobileNet. Загалом, у всіх випадках була застосована та сама активаційна функція (сигмоїд), аналогічні налаштування нормалізації, кількість нейронів у щільних шарах (1 нейрон для завдання бінарної класифікації), та оптимізаційна функція (Adam) та функцію втрати (BCEWithLogitsLoss).

Налаштування доповнення даних у тих прототипах, де ця техніка була застосована була ідентичною та включала перевертання 50% зображень, однаковий розмір зображень, перевертання зображень випадковим чином на 15 градусів, зміна яскравості зображень, та подекуди була використана техніка випадкового обрізання зображення.

Основні експерименти проводилися з показниками швидкості навчання, розмір пакета, доповнення даних та налаштування замороження шарів моделі.

У перших чотирьох експериментах доповнення даних не застосовувалось, провели експерименти з різними розмірами пакета (10, 32 та 50 записів у ітерації) та швидкостями навчання (0.01 та 0.001) – проте точність всіх цих спроб на основі ПНМ MobileNet досягла 31%. У п'ятому

випадку додається доповнення даних, і точність передбачення зросла до 38%.

Наступні п'ять експериментів використовували техніку замороження шарів, зокрема також були проведені експерименти з різними пропорціями замороження шарів. Результати останніх двох експериментів досягають найвищих результатів точності у 92%. У цих експериментах використані найменший серед усіх показник швидкості навчання – 0.0001, та найменший розмір пакета серед усіх експериментів, що становить десять.

Також відмінність відбувалась у кількості епох, що були збільшені від 11 до 20. При цьому експеримент з налаштуваннями як у дев'ятому експерименті використовує 30% заморожених шарів, а десятий – 70%, але результат точності однаковий. На рисунку 3.2 зображений класифікаційний звіт роботи класифікації десятого експерименту, результати якого є ідентичними до дев'ятого та одинадцятого експериментів.

Classification Report:				
	precision	recall	f1-score	support
European City	0.90	1.00	0.95	9
American City	1.00	0.75	0.86	4
accuracy			0.92	13
macro avg	0.95	0.88	0.90	13
weighted avg	0.93	0.92	0.92	13

Рисунок 3.2 – Класифікаційний звіт десятого експерименту на основі ПНМ MobileNet

При відтворенні цих перших десяти експериментів збалансування класів не було застосовано. Й тому, відповідно клас з меншими кількостями значення у тесті мають нижчий рівень передбачення у порівнянні з класом де майже вдвічі більше значень.

У 11-му експерименті застосовується техніка присвоєння ваги для класів з метою збалансування результатів; таке рішення було прийнято через невеликий об'єм вибірки. Були проведені різні варіанти застосування ваг: зокрема автоматичне та ручне налаштування, проте у всіх випадках результати залишались попередніми та не вдосконалювались.

Найвищі показники результатів бінарної класифікації для першого завдання були виявлені при використанні моделі з налаштуваннями як у дев'ятому, десятому та одинадцятому випадках. Модель з налаштуваннями як у цих експериментах досягають однакового рівня точности – 92%. Експерименти характеризуються застосуванням замороження шарів у різних пропорціях (30%, 50% та 70% були використані відповідно у 9-му, 10-му та 11-му експериментах), використання малих розмірів пакета (у всіх трьох по 10 зображень), використання техніки доповнення даних, модель навчалась за 20 епох, та був використаний найнижчий (серед усіх проведених експериментів) індекс швидкості навчання (складав 0.0001). Приклад використаної ПНМ MobileNet представлена у додатку А. На рисунку 3.3 представлені графік точности та графік втрати у процесі навчання моделі.

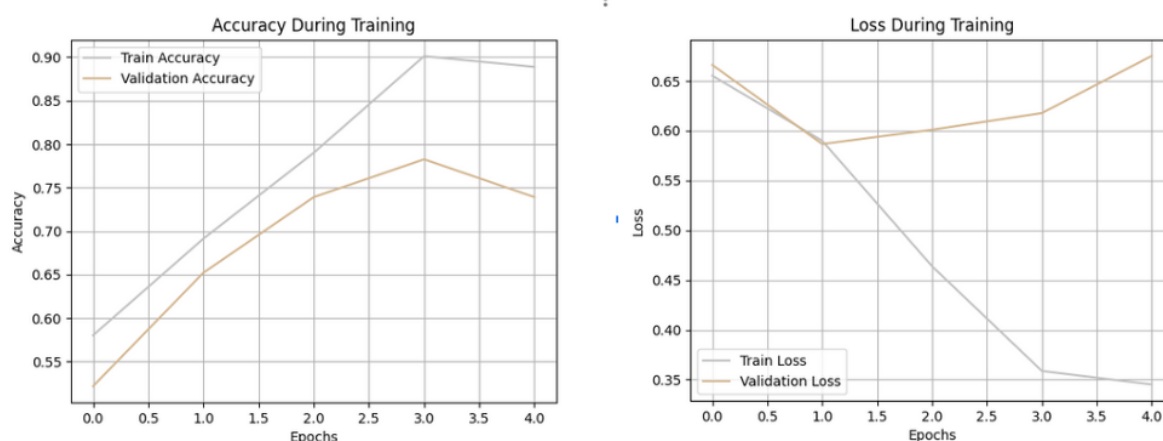


Рисунок 3.3 – Графік точности та втрати десятого експерименту на основі ПНМ MobileNet

Для розв'язання першого завдання бінарної класифікації з використанням ПНМ EfficientNet було відтворено десять експериментів з різними налаштуваннями. Загалом, результати класифікації були не стійкими та помірними.

Перші чотири експерименти не використовували збалансування (оскільки перше завдання мало у розпорядженні дуже малу вибірку, підтримка кожного класу під час тестування становила 9 до 4, тобто один клас переважав більш ніж вдвічі). Перших два експерименти були зосереджені на швидкості навчання. При цьому обидві версії не використовували доповнення даних та замороження шарів.

Повільніше навчання показало вищу ефективність у класифікації: налаштування першого експерименту містило швидкість навчання 0.001, і при цьому класифікувала правильно 100% американських міст та 22% історичних; швидкість навчання налаштування другого експерименту знижується до 0.0001, і з такими параметрами модель досягає вже 69% точності й ситуація з передбаченням європейських міст набагато покращується.

Найгірші результати роботи моделі були при таких її характеристиках: 30% заморожених шарів, розмір пакета – 70 записів, використання додавання даних, швидкість навчання становила 0.001. При таких налаштуваннях точність моделі була 38%. На рисунку 3.4 представлений класифікаційний звіт десятого експерименту на основі ПНМ EfficientNet, який мав найгірший результат класифікації серед усіх експериментів у цій категорії.

Приклад використаної ПНМ EfficientNet для вирішення як першого, так і другого завдання представлена у додатку Б.

Найліпшого результату модель на основі EfficientNet досягла при налаштуваннях як у третьому експерименті: замороження шарів та доповнення даних не були застосовані, розмір пакета у 50, 11 епох та

повільне навчання (0.0001). Модель з описаними параметрами досягла 85% точности.

Classification Report:				
	precision	recall	f1-score	support
European City	1.00	0.11	0.20	9
American City	0.33	1.00	0.50	4
accuracy			0.38	13
macro avg	0.67	0.56	0.35	13
weighted avg	0.79	0.38	0.29	13

Рисунок 3.4 – Класифікаційний звіт десятого експерименту на основі ПНМ EfficientNet

З метою виконання першого завдання на основі ПНМ ResNet було проведено чотири експерименти. Загалом, результати класифікації на основі точности були невтішними. Точність коливалась у межах від 49% до 69%. Були проведені експерименти на основі змін швидкості навчання, розміру пакета, терпіння ранньої зупинки та методів регуляризації. Зокрема, були використанню техніки доповнення даних, використання індексу терпіння ранньої зупинки, та налаштування гіперпараметрів, які могли б запобігати перенавчанню. В останньому експерименті був також застосований метод регуляризації Dropout, де $p = 50\%$.

Проте значних успіхів у класифікації з використанням цієї ПНМ досягти не вдалось. Така модель дійсно не є найкращим варіантом у випадку малих об'ємах вибірки.

Приклад використаної ПНМ ResNet для вирішення обох завдань на презентований у додатку В.

3.2.2 Розв'язання другого завдання

Для розв'язання другого завдання на основі ПНМ MobileNet було проведено шість експериментів з різними налаштуваннями для досягнення найвищого рівня точності.

Оскільки, зображення збирались автоматизовано, у вибірці подекуди були важкі зображення, які необхідно було відфільтрувати, що є відмінністю від першого завдання. Таким чином, вибірка сильно зменшилась, а клас сучасних мап зменшитись від шести тисяч зображень до чотирьох тисяч. Тому кількість всіх зображень у всій вибірці впала від 12 тисяч до менш як 8 тисяч зображень.

Загалом, були використані налаштування моделі, які показали високу точність для розв'язання першого завдання, що характеризувались невеликим розміром пакета, низьким значенням швидкості навчання та застосуванням техніки доповнення даних.

Модель першого експерименту для цього завдання мала наступні параметри: 70% замороження шарів, пакет містив десять записів, доповнення даних не застосовувалось, навчання відбувалось впродовж 20 епох, та низьке значення швидкості навчання (0.0001).

Наступні два експерименти характеризувались пришвидшення навчання (до рівня 0.1) та зменшення епох (до п'яти).

У п'ятому експерименті не була використана техніка замороження шарів на відмінну від усіх інших п'яти експериментах. Попри усі зміни у налаштуваннях моделі рівень точності залишався незмінним та досягав 93%.

У останньому експерименті модель досягає точности у 95%, що є найкращим результатом серед проведених експериментів на основі MobileNet для класифікації історичних та сучасних мап. На рисунку 3.5 презентований класифікаційний звіт роботи шостого експерименту, результати класифікації якого є для розв'язку цього завдання.

Classification Report:				
	precision	recall	f1-score	support
modern map	0.97	0.93	0.95	419
historical map	0.91	0.97	0.94	293
accuracy			0.95	712
macro avg	0.94	0.95	0.94	712
weighted avg	0.95	0.95	0.95	712

Рисунок 3.5 – Звіт класифікації для шостого експерименту (MobileNet для вирішення класифікації історичних та сучасних мап)

Аналіз звітів класифікації моделі, вказує на те, що модель визначає правильно більше сучасних мап. У шістьох експериментах точність позитивних передбачень коливається в межах 96%–99%, тоді як для історичних мап цей показник менший та коливається від 86% та 91% (91% точність позитивних передбачень зустрічається саме у шостому експерименті, який має найбільшу точність передбачень серед усіх моделей). Це вказує на те, що модель, подекуди, визначала деякі сучасні мапи історичними.

Найкращі результати модель досягла при найменшій швидкості навчання (з обраних для експериментів), з розміром пакета у 32 записи, з 30% заморожених шарів та без доповнення даних. Подібні налаштування є у четвертому експерименті, проте у налаштуваннях цієї моделі розмір пакета був майже втричі менший, та модель досягала 93% точності. Графіки точності та втрати у процесі класифікації можна переглянути на рисунку 3.6.

Для розв’язання другої задачі з використанням ПНМ EfficientNet було відтворено шість експериментів з різними параметрами. Всі класи моделі були зважені, домінантному класу була надана вага 0.3.

Перші три експерименти мали на увазі випробування з розміром пакета: були випробувана модель з 10, 70 та 100 записами.

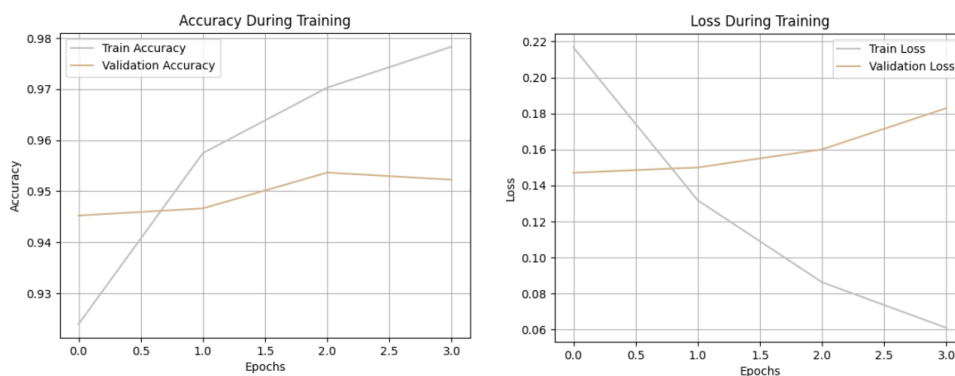


Рисунок 3.6 – Графіки точності та втрати шостого експерименту моделі на основі MobileNet

Всі ці варіанти використовували доповнення даних, та мали 30% заморожених шарів, та мали однаковий результат точності – 94%.

Проте зі збільшенням розміру пакета, збільшується точність позитивних передбачень для першого класу (сучасні мапи) від 89% у першому експерименті до 90% у третьому. 95% точності класифікації досяг п'ятий експеримент, модель якого мала наступні характеристики: 50% заморожених шарів, використання доповнених даних, 100 записів у пакеті, швидкість навчання, що дорівнювала 0.001. Два експерименти досягли результату у 96% точності: четвертий (30% заморожених шарів, 100 записів у пакеті, швидкість навчання становила 0.001, без доповнення даних) та шостий (відмінними від четвертого експерименту у налаштуваннях є тільки не використання замороження шарів). Різниця у даних з класифікаційних звітах обох цих експериментів не було виявлено.

Для розв'язку другого завдання на основі ПНМ ResNet було проведено десять експериментів. Параметри перших чотирьох експериментів використовували техніку замороження шарів, у пропорціях від 30% до 50%, та не використовували доповнення даних. Всі чотири випадки, мали низький індекс швидкості навчання (0.0001), та досягли точності від 95% до 96%. Зокрема, саме збільшення пропорції замороження шарів, досягло точності 96%. П'ятий експеримент характеризувався

додаванням доповнення даних, що дало нижчий результат у порівнянні з попередніми – 93%. Шостий та сьомий експерименти не використовували замороження шарів, проте з використанням доповнених даних, та експериментами зі швидкістю навчання, що показало, що нижча швидкість навчання у цих конкретних налаштуваннях, приводить до кращої класифікації. Восьмий та дев'ятий експерименти не використовували ні доповнення даних, ні замороження, а самі зміни базувались на швидкості навчання. Результати цих двох експериментів вказали, що доцільно надавати низький індекс швидкості навчання, адже саме дев'ятий експеримент зі вказаною швидкістю 0.01, мала найнижчі результати класифікації цього завдання на основі моделі ResNet – 87%. Останній експеримент також не показав цікавих результатів.

Узагальнюючи, використання ResNet показує ефективність при наявності великої вибірки для класифікації, а найкращих результатів модель досягає при налаштуваннях як у четвертому експерименті для вирішення другого завдання.

Налаштування моделей, а також результати класифікації (тільки значення точності класифікації) для всіх експериментів презентовані у додатку Г.

3.3 Розробка системи

Система, що розробляється у рамках роботи включає модель передбачення (дві обрані моделі, які показали найвищі результати класифікації в експериментах попередньої частини роботи), інтерфейс (фронт-енд) та бекенд.

Для побудови системи були використані: HTML для написання каркаса основної сторінки, CSS для створення стилів та JavaScript для створення динамічних ефектів. Для бекенду була використана мова програмування Python, так само як і для створення моделей класифікації.

3.3.1 Розробка UI/UX

До практичної реалізації та побудови користувацького інтерфейсу, відбулось дослідження систем, що пропонують користувачам різноманітні послуги пов'язані з зображеннями. До таких операцій можемо віднести: зменшення зображення у розмірі, перетворення одного формату зображення на інше тощо.

Було проаналізовано інтерфейси п'ятих систем: compresssystem.io, products.aspose.app, iloveimg.com, png2pdf.com, ilovepdf.com.

Головна сторінка першої системи містить анімоване зображення з правої сторони та поле для завантаження зображення зліва; головна мета це стискання зображення у форматі png та jpg; нижче представлені переваги використання саме цієї системи, пояснення, що таке стискання зображення та навіщо це робити, і порівняння з конкурентами; при завантаженні зображення стискання відбувається автоматично, і користувач має натиснути кнопку завантаження на комп'ютер вже стисненого зображення.

Друга система (products.aspose.app) пропонує ряд операцій (стискання, пошук об'єктів на зображенні, стирання водяного знаку, обрізання тощо), доступ до яких розміщений у компактному меню і вигляді стовпців з лівою сторони сторінки. Після того, як користувач обирає операцію, завантажує картинку, користувач має натиснути на кнопку аби розпочати процес операції; ця система також надає більше мобільності у налаштуваннях.

Третя система також надає ряд різноманітних послуг, варіанти доступні користувачу у верхньому стрічковому меню. Після вибору операції, користувач переадресовується на нову сторінку сайту, де має завантажити картинку та натиснути на кнопку для початку роботи, опісля того як процес завершився, зображення автоматично завантажується на комп'ютер, при цьому також доступна кнопка для повторного завантаження.

Четверта система, а саме png2pdf.com також пропонує різні операції (більшість пов'язана з форматом pdf), при зміні будь-якої з них, перезавантажується система, проте користувач залишається на головній сторінці; користувач має завантажити картинку та натиснути на кнопку для початку роботи, опісля – користувач знову має натиснути на кнопку для завантаження перетвореного файлу.

Остання система (ilovepdf.com), у якій користувач має обрати операцію, після чого, він чи вона будуть перенаправлені на іншу сторінку, де користувач має завантажити зображення, потім користувач може обрати певні налаштування (наприклад, вертикальний чи горизонтальний формат) зображення, після чого користувач натикає на кнопку для початку операції. Коли операція завершена, файл автоматично завантажено та додатково є кнопка для ручного завантаження.

На основі аналізу вищезгаданих платформ виводимо висновки:

– якщо система надає можливості редагування та налаштування фінального результату, необхідність додаткової кнопки для початку роботи над зображенням є виправданою, проте якщо додаткові налаштування відсутні, ця функція займає час у користувача;

– у тому разі, якщо система є мультифункціональною, необхідно зробити розділ, певне меню з переліком можливих операцій для розуміння користувача.

Побудована діаграма послідовностей (рисунок 3.7) вказує на частоту взаємодії між об'єктами та суб'єктами системи, та практично може допомогти знизити тиск чи кількість запитів між об'єктами в один і той самий проміжок часу.

Також був розроблений макет користувацького інтерфейсу (рисунок 3.8) для підбору правильного зонування секцій та визначення необхідних елементів та розділів.

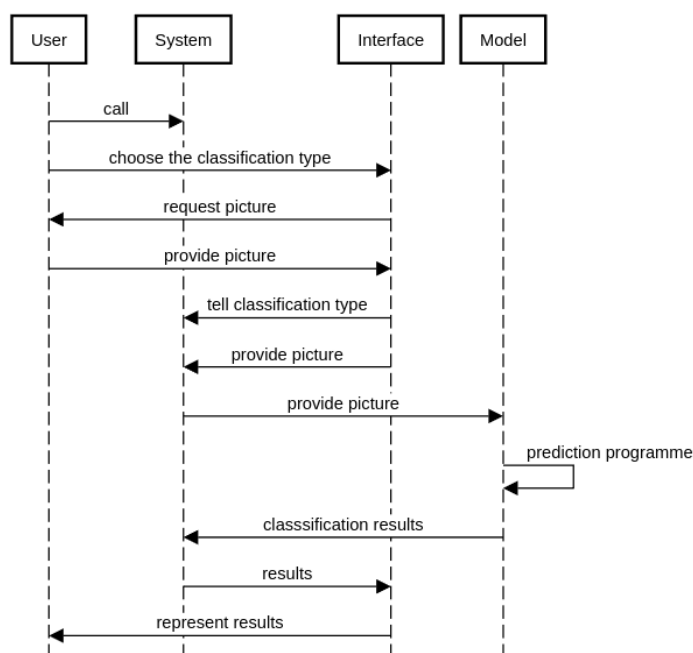


Рисунок 3.7 – Діаграма послідовностей та взаємодії об'єктів

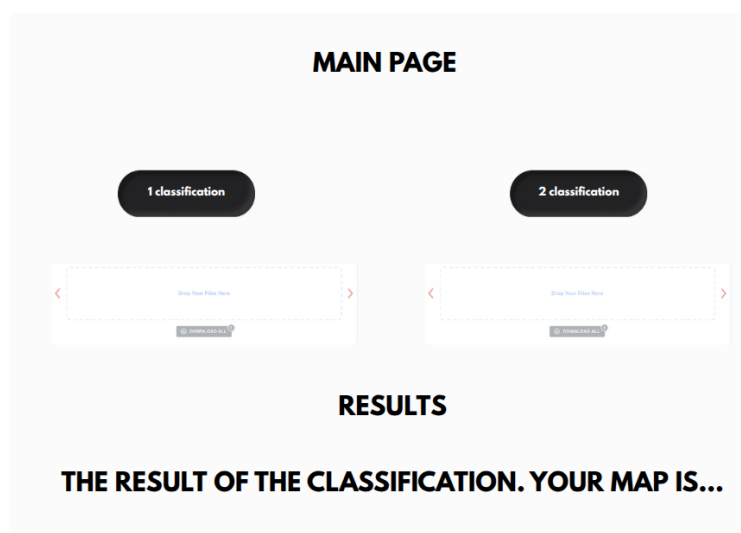


Рисунок 3.8 – Макет користувацького інтерфейсу

Причина вибору односторінкового сайту це збереження часу користувача, який не чекатиме на завантаження сторінки, особливо якщо сайт складається з великої кількості сторінок. За логікою, система буде передбачати клас зображення на основі вибору поля, куди світлина була завантажена. Аналогічно, результати передбачення не будуть відкриватись

додатково на новій сторінці, куди користувач має бути перенаправлений, а з'являються внизу тієї самої сторінки.

Фінальна версія користувацького інтерфейсу повторює схему попередньо розробленого макета інтерфейсу. На горіщі сайту бачимо заголовок, що вказує користувачу основну мету інструменту, дві горизонтальні лінії візуально розділяють сторінку на три зони: верхня з назвою вказує на мету; друга частина – робоча, де користувач обирає тип класифікації, завантажує зображення та нижче отримує результат передбачення. Та остання третя зона, де розміщені інструкції користування.

Інструкція розміщена внизу сторінки з метою звільнення простору для користувачів, які інтуїтивно розуміють принципи взаємодії з застосунком. Палетка, за якою були застосовані кольори містила наступні відтінки: #FCE7C8, #B1C29E, #FADA7A, #F0A04B.

Остаточний вигляд користувацького інтерфейсу презентований на рисунку 3.9.



Рисунок 3.9 – Результат розробки користувацького інтерфейсу

Додатково, коли користувач розміщує курсор у секції двох типів класифікації, секція підсвічується. Вікно зліва змінює колір на #FADA7A, а секція справа на #F0A04B. Такий ефект було додано, для більш ефективної

візуальної орієнтації користувача та задля зменшення помилки у виборі типу класифікації.

3.3.2 Розробка бекенду

Розробка бекенду почалась з етапу адаптації моделей класифікації, для того, аби була можливість класифікувати тільки одне зображення за одну сесію.

Для розв'язання першого завдання, була обрана дев'ята модель на основі MobileNetV2, яка досягла найвищих результатів класифікації. Тоді як для другого завдання була обрана модель на основі EfficientNetB0.

Оскільки для створення системи був використаний новий проєкт, обидві моделі, а також pth файли до них (у яких збереження налаштування моделі, такі як використання замороження шарів, швидкість навчання тощо), що зберігаються ваги моделі були скопійовані до нового проєкту.

Структура проєкту system наведена на рисунку 3.10.

```
/system
├── __pycache__
├── model
│   ├── __pycache__
│   ├── __init__.py
│   ├── model1.py
│   ├── model1.pth
│   ├── model2.py
│   └── model2.pth
├── venv
├── .dockerignore
├── Docketfile
├── index.html
├── main.py
├── requirements.txt
├── script.js
└── styles.css
```

Рисунок 3.9 – Структура проєкту system

Частина файлів у структурі проєкту належать до контейнеризації (власне, Dockerfile та .dockerignore – для ігнорування віртуального середовища), а деякі саме з віртуальним середовищем проєкту.

Опісля всі файли розробленої системи були завантажені на GitHub.

3.4 Перспективи розвитку системи

Розроблена система має ряд недоліків та зон, які потенційно можливо вдосконалити.

Для завдання класифікації обрисів міст з американських континентів та європейської частини Євразії (перше завдання):

- для ефективного навчання та результатів, необхідно суттєво збільшити вибірку;
- додавання інших класів – міста з інших частин світу та застосування багатокласової класифікації.

Для другого завдання покращенням є використання мультикласової класифікації з класами по десятиліттю створення мапи. Така система, може бути використана у навчальних закладах з метою освіти, або для протидії пропаганди з метою визначення неправдивої інформації.

Додача зображень мап для інтерфейсу, з метою інтуїтивного користування, розробка додаткової сторінки, де будуть описуватись результати моделей. Також надати змогу користувача самостійно коригувати параметри ПНМ для класифікації необхідного зображення.

ВИСНОВКИ

У результаті роботи над роботою було:

– досліджено сучасні знахідки у галузі комп'ютерного зору, сучасні виклики пов'язані з цим напрямком, а також потенційне практичне застосування цих технологій; досліджено як наразі технології використовуються, в тому числі класифікація зображення у комплексі з іншими задачами комп'ютерного зору;

– було проведено дослідження ПНМ, а саме як проєктується побудова таких моделей, яка їхня мета, переваги та недоліки для класифікації зображень, у тому числі картографічних матеріалів;

– було проведено 74 експерименти з трьома різними ПНМ для розв'язання двох задач класифікації, задля дослідження параметрів, за яких класифікація є найбільш точною;

– досліджено онлайн вебзастосунки, що функціонують та пропонують свої послуги користувач по роботі з зображеннями, а на основі аналізу цих застосунків, розроблений шлях користувача для системи, що була розроблена;

– був розроблений бекенд, в тому числі модифікація двох моделей, які показали найвищі результати класифікації для обох завдань, перенесення самих моделей та їх розроблених ваг, а опісля відбувалась контейнеризація.

У процесі роботи над кваліфікаційною роботою були виконані усі поставленні задачі, та досягнути мета дослідження.

У результаті експериментів над ПНМ були виокремлені дві моделі на основі MobileNet для першої вибірки, яка мала доволі малий об'єм, через що багато моделей не спрацювали добре, що було передбачено у деяких випадках аналізом теоретичних та наукових напрацювань досліджених у першій частині роботи.

Для розв'язання другого завдання модель на основі EfficientNet, результати якої перевершили результати інших експериментів на основі цієї моделі.

Розроблена система вимагає ряд вдосконалень, особливо у контексті комерційної реалізації, та потенційно може бути використана з метою визначення різних періодів мап, для визначення типу мапи, континенту або у процесі перевірки фактів з метою визначення ступеня правдивості, а також з метою дослідження урбаністики та покращення розвитку міст.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. History of computer vision – let's data science. *Let's Data Science*. URL: <https://letsdatascience.com/learn/history/history-of-computer-vision/> (date of access: 05.06.2025).
2. GeeksforGeeks. Comprehensive guide to edge detection algorithms – Geeksforgeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/comprehensive-guide-to-edge-detection-algorithms/> (date of access: 05.06.2025).
3. Ambika. What is computer vision? (history, applications, challenges). *Medium*. URL: <https://medium.com/@ambika199820/what-is-computer-vision-history-applications-challenges-13f5759b48a5> (date of access: 05.06.2025).
4. Bhuyan M. K. Computer vision and image processing: fundamentals and applications. CRC Press, 2019. 468 c.
5. Ivankov A. Tasks and applications of computer vision – profolus. *Profolus*. URL: <https://www.profolus.com/topics/tasks-and-applications-of-computer-vision/> (date of access: 05.06.2025).
6. Patel H. 15 computer visions projects you can do right now. *neptune.ai*. URL: <https://neptune.ai/blog/15-computer-visions-projects> (date of access: 05.06.2025).
7. Boesch G. Computer vision tasks (comprehensive 2025 guide) – viso.ai. *viso.ai*. URL: <https://viso.ai/deep-learning/computer-vision-tasks/> (date of access: 05.06.2025).
8. 8 real-world applications of computer vision in 2023. *Augmented AI*. URL: <https://www.augmentedstartups.com/blog/8-real-world-applications-of-computer-vision-a-comprehensive-guide> (date of access: 05.06.2025).
9. Good O. How Google Translate squeezes deep learning onto a phone. *Google Research*. URL: <https://research.google/blog/how-google-translate-squeezes-deep-learning-onto-a-phone/> (date of access: 05.06.2025).

10. 5 common problems with computer vision and their solutions. *Chooch*. URL: <https://www.chooch.com/blog/5-common-problems-with-computer-vision-and-their-solutions/> (date of access: 05.06.2025).
11. Bharat. Your 2025 guide to the top 6 computer vision problems. *OpenCV*. URL: <https://opencv.org/blog/computer-vision-problems/> (date of access: 05.06.2025).
12. Goodfellow I., Bengio Y., Courville A. Deep learning. Cambridge : The MIT Press, 2015. 800 c.
13. Pujari P., Sewak M., Karim R. Practical convolutional neural network models. Packt Publishing, 2018. 218 c.
14. Devjak. Unlocking the secrets of instagram's personalized recommendations. *Medium*. URL: <https://devjacks.medium.com/unlocking-the-secrets-of-instagrams-personalized-recommendations-a012ddcfbd30> (date of access: 05.06.2025).
15. One R. Exploring the future of visual search: an in-depth guide to google lens architecture and technology. *Medium*. URL: <https://medium.com/@rc-one/exploring-the-future-of-visual-search-an-in-depth-guide-to-google-lens-architecture-and-technology-6c5e83357f00> (date of access: 05.06.2025).
16. Boesch G. Vision transformers (vit) in image recognition. *viso.ai*. URL: [https://viso.ai/deep-learning/vision-transformer-vit/#:~:text=Vision%20Transformers%20\(ViT\)%20is%20an,and%20a%20feed-forward%20layer](https://viso.ai/deep-learning/vision-transformer-vit/#:~:text=Vision%20Transformers%20(ViT)%20is%20an,and%20a%20feed-forward%20layer) (date of access: 05.06.2025).
17. Khan Suri Z. Convolution vs. attention. *Curiosity*. URL: <https://zshn25.github.io/CNNs-vs-Transformers/> (date of access: 05.06.2025).
18. Duodu J. Attention mechanism and vision transformer. *Medium*. URL: <https://medium.com/@jduodu5/attention-mechanism-and-vision-transformer-8955f93c53d2> (date of access: 05.06.2025).

19. van der Werff T. CNN vs. vision transformer: a practitioner's guide to selecting the right model. *Tobias van der Werff*. URL: <https://tobiasvanderwerff.com/2024/05/15/cnn-vs-vit.html> (date of access: 05.06.2025).

20. Mwit D. Transfer learning guide: a practical tutorial with examples for images and text in keras. *neptune.ai*. URL: <https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras> (date of access: 05.06.2025).

21. AI Brilliance. Transfer learning and pre-trained models. *LinkedIn*. URL: <https://www.linkedin.com/pulse/transfer-learning-pre-trained-models-ai-brilliance-4hwcc/> (date of access: 05.06.2025).

22. Karzhev S. Overview of advanced transfer learning techniques. *DataCamp*. URL: <https://www.datacamp.com/blog/transfer-learning-advanced> (date of access: 05.06.2025).

23. Pre trained model. *Encord*. URL: <https://encord.com/glossary/pre-trained-model-definition/> (date of access: 05.06.2025).

24. Сорока Є. Попередньо навчені моделі. *r_d media*. URL: <https://robotdreams.cc/uk/blog/115-predobuchennye-modeli> (дата звернення: 05.06.2025).

25. Ahammad M., Ali F., Sachdeva M. What are some of the most popular and widely used pre-trained models for deep learning?. *LinkedIn*. URL: <https://www.linkedin.com/advice/0/what-some-most-popular-widely-used-pre-trained> (date of access: 05.06.2025).

26. MobileNet V2. *HuggingFace*. URL: https://huggingface.co/docs/transformers/model_doc/mobilenet_v2 (date of access: 05.06.2025).

27. Kirouane A. MobileNet-V2 vs MobileNet-V3. *LinkedIn*. URL: <https://www.linkedin.com/pulse/mobilenet-v2-vs-mobilenet-v3-ayoub-kirouane/> (date of access: 05.06.2025).

28. Tan M., Le Q. V. EfficientNet: rethinking model scaling for convolutional neural networks. *ArXiv*. 2020. T. 1905.11946. URL: <https://arxiv.org/abs/1905.11946> (date of access: 05.06.2025).

29. Sarkar A. Understanding EfficientNet – The most powerful CNN architecture. *Medium*. URL: <https://arjun-sarkar786.medium.com/understanding-efficientnet-the-most-powerful-cnn-architecture-eaeb40386fad> (date of access: 05.06.2025).

30. Tashin A., Nuri Sabab N. H. Classification and understanding of cloud structures via satellite images with EfficientUNet. *SN computer science*. 2021. T. 3, № 1. URL: <https://essopenarchive.org/users/530877/articles/597477-classification-and-understanding-of-cloud-structures-via-satellite-images-with-efficientunet> (date of access: 05.06.2025).

31. ResNet. *HuggingFace*. URL: https://huggingface.co/docs/transformers/en/model_doc/resnet (date of access: 05.06.2025).

32. Deep residual learning for image recognition / K. He *Arxiv*. 2015. T. 1512.03385. URL: <https://databasecamp.de/en/ml/resnet-en> (date of access: 05.06.2025).

33. ResNet: residual neural networks – easily explained! | data basecamp. *Data Basecamp*. URL: <https://databasecamp.de/en/ml/resnet-en> (date of access: 05.06.2025).

34. Pretrained deep learning models. *Esri*. URL: <https://www.esri.com/en-us/arcgis/deep-learning-models> (date of access: 05.06.2025).

35. What are the consequences of not freezing layers in the transfer learning?. *StackExchange*. URL: <https://datascience.stackexchange.com/questions/30659/what-are-the-consequences-of-not-freezing-layers-in-transfer-learning> (date of access: 05.06.2025).

36. Fazla A., Guzman Piedrahita D., Deo Manekar T. Exploring freezing in transfer learning. *Department of Infomatics of University of Zurich*.

URL: https://www.ifi.uzh.ch/dam/jcr:76accb6-0dfd-4b5c-9df8-ee049133f9f9/fazla_guzman_manekar2024master_project.pdf (date of access: 05.06.2025).

What is data augmentation?. *Amazon Web Services.*

URL: [https://aws.amazon.com/what-is/data-augmentation/#:~:text=Data%20augmentation%20is%20the%20process,machine%20learning%20\(ML\)%20models](https://aws.amazon.com/what-is/data-augmentation/#:~:text=Data%20augmentation%20is%20the%20process,machine%20learning%20(ML)%20models) (date of access: 05.06.2025).