

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра комп'ютерних наук**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»**

**ДОСЛІДЖЕННЯ ТА АНАЛІЗ СТАНУ ЗДОРОВ'Я ПАЦІЄНТІВ ЗА
ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ**

**RESEARCH AND ANALYSIS OF PATIENT HEALTH CONDITIONS USING
ARTIFICIAL INTELLIGENCE**

спеціальність 122 «Комп'ютерні науки»

освітня програма «Комп'ютерні науки»

Виконав: здобувач вищої освіти
групи КНм-21
Зданевич Ярослав Ігорович

(підпис)

Керівник: к.т.н., доцент
Лук'янчук Юрій Анатолійович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« » _____ 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Валерій Олександрович

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерних наук

Ступінь вищої освіти: магістр

Галузь знань: 12 Інформаційні технології

Спеціальність: 122 Комп'ютерні науки

Освітня програма: «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Валерій ЛПЦИНА

«14» травня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ДРУГОГО (МАГІСТЕРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ

Зданевич Ярослав Ігорович

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Дослідження та аналіз стану здоров'я пацієнтів за допомогою штучного інтелекту»

Керівник роботи к.т.н., доцент Лук'ячук Юрій Анатолійович

затверджені наказом закладу вищої освіти від «14» травня 2025 р. № 255/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «15» грудня 2025 р.

3. Вихідні дані до роботи: статті, дослідження вітчизняних та закордонних авторів в даній області, сучасні методи і засоби розробки, технічна документація технологій розробки.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): Аналіз проблематики та постановка завдань дослідження, теоретичне дослідження та практична реалізація інтелектуальної системи аналізу рентгенівських знімків легень на основі глибокого навчання, експериментальне дослідження результативності системи аналізу рентгенівських знімків легень на основі методів глибокого навчання.

5. Перелік графічного матеріалу: 1. Схема принципу роботи згорткової нейронної мережі (CNN); 2. Порівняльний аналіз сучасних систем автоматизованої діагностики рентгенівських знімків; 3. Узагальнена схема застосування глибокого навчання в медичній візуалізації; 4. Архітектура системи MediVision; 5. Архітектура нейронної мережі MediVision (MobileNetV2 + Dense); 6. Послідовність виконання інференсу моделі MediVision; 7. UML-діаграма класів системи інференсу; 8. Діаграма варіантів використання (Use Case) системи MediVision; 9. Динаміка функції втрат (loss) під час навчання; 10. Динаміка зміни AUC-ROC; 11. Динаміка метрики AUC-PR; 12. Різниця між train та validation AUC-PR; 13. Динаміка зміни Binary Accuracy; 14. Таблиця характеристик використаних наборів даних; 15. Таблиця узагальнених показників точності класифікації патологій.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблематики та постановка завдань дослідження</i>	<i>Лук'янчук Ю. А.</i>		
<i>Теоретичне дослідження та практична реалізація системи розпізнавання патолгій легень на основі ШІ</i>	<i>Лук'янчук Ю. А.</i>		
<i>Експериментальне дослідження результативності системи розпізнавання патолгій легень на основі ШІ</i>	<i>Лук'янчук Ю. А.</i>		
<i>Показник запозичень тексту</i>		%	
<i>Інструментальна перевірка</i>	<i>Кошелюк В. А.</i>		
<i>Нормоконтроль</i>	<i>Сачук В. О.</i>		
<i>Гарант ОПП</i>	<i>Ліщина В. О.</i>		

7. Дата видачі завдання «14» травня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи магістра	Строк виконання етапів роботи	Примітка
1.	<i>Обґрунтування теми дослідження</i>	<i>до 14.05.2025</i>	
2.	<i>Провести огляд літературних джерел по темі кваліфікаційної роботи</i>	<i>до 21.08.2025</i>	
3.	<i>Провести аналіз загальної проблеми і вибір напрямків дослідження</i>	<i>до 05.09.2025</i>	
4.	<i>Розробити функціональну схему роботи програмного продукту</i>	<i>до 30.09.2025</i>	
5.	<i>Описати засоби розробки об'єкта проектування</i>	<i>до 09.10.2025</i>	
6.	<i>Практична реалізація об'єкта проектування</i>	<i>до 20.10.2025</i>	
7.	<i>Розробити методіку для проведення експерименту</i>	<i>до 29.10.2025</i>	
8.	<i>Провести аналіз результатів експерименту</i>	<i>до 12.11.2025</i>	
9.	<i>Формування списку використаних джерел</i>	<i>до 17.11.2025</i>	
10.	<i>Оформлення ілюстративного матеріалу</i>	<i>до 25.11.2025</i>	
11.	<i>Інструментальна перевірка на академічний плагіат</i>	<i>до 03.12.2025</i>	
12.	<i>Здача чистового варіанту кваліфікаційної роботи на кафедрі</i>	<i>до 05.12.2025</i>	

Здобувач вищої освіти _____ Ярослав ЗДАНЕВИЧ

Керівник роботи _____ Юрій ЛУК'ЯНЧУК

АНОТАЦІЯ

Зданевич Я. І. Дослідження та аналіз стану здоров'я пацієнтів за рентгенівськими зображеннями легень із використанням методів глибокого навчання. Рукопис.

Кваліфікаційна робота магістра ОП «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота магістра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків (згідно зі структурою, затвердженою кафедрою).

У роботі досліджено сучасні методи глибокого навчання, орієнтовані на автоматизований аналіз рентгенівських знімків грудної клітки з метою виявлення патологій органів дихання. Розглянуто підходи до попередньої обробки медичних зображень, застосування згорткових нейронних мереж та особливості мульти-label класифікації.

Розроблено програмний прототип системи MediVision, який базується на архітектурі MobileNetV2 та реалізує повний цикл аналізу зображень: підготовку даних, навчання моделі, оптимізацію порогів класифікації й модуль інференсу для прогнозування стану пацієнта за одним знімком. Проведено тренування моделі, оцінювання її якості та аналіз результатів за ключовими метриками – AUC-ROC, AUC-PR, BinaryAccuracy.

Експериментальні результати засвідчують високу точність класифікації, стабільність роботи. Окреслено перспективи подальшого розвитку системи, включно з розширенням наборів даних та вдосконаленням методів пояснюваності результатів.

Ключові слова: медична діагностика, глибоке навчання, рентгенографія легень, мультикласова класифікація, MobileNetV2, штучний інтелект, MediVision.

ABSTRACT

Yaroslav Zdanevych. Research and analysis of patients' health conditions based on chest X-ray images using deep learning methods. Manuscript.

Qualification work for master's degree in «Computer Science» specialty 122 «Computer Science». Lutsk National Technical University. Lutsk, 2025.

The master's thesis consists of an introduction, three chapters, conclusions and proposals, a list of references, and appendices in accordance with the structure approved by the department.

This work investigates modern deep learning methods applied to the automated analysis of chest X-ray images for detecting pulmonary pathologies. The study examines approaches to medical image preprocessing, convolutional neural networks, and the specifics of multi-label classification. Neural network architectures and their effectiveness in medical diagnostic tasks are analyzed.

A software prototype of the MediVision system was developed based on the MobileNetV2 architecture, implementing a full image analysis pipeline: data preprocessing, model training, threshold optimization, and an inference module for predicting a patient's condition from a single image. The model was trained, evaluated, and analyzed using key metrics, including AUC-ROC, AUC-PR, and Binary Accuracy.

Experimental results demonstrate high classification accuracy, model stability, and practical suitability of MediVision as a tool for preliminary diagnostics in medical institutions. The thesis outlines further development prospects, including dataset expansion and the enhancement of explainability methods.

Keywords: medical diagnostics, deep learning, chest X-ray analysis, multi-label classification, MobileNetV2, artificial intelligence, MediVision.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ	10
1.1 Огляд і аналіз предметної області проблеми (задачі), результатів існуючих теоретичних та експериментальних досліджень	10
1.2 Огляд і аналіз методів та засобів розробки системи розпізнавання патологій легень на основі штучного інтелекту для вирішення проблеми дослідження....	13
1.3 Постановка завдання на кваліфікаційну роботу магістра.....	16
РОЗДІЛ 2 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ АНАЛІЗУ РЕНТГЕНІВСЬКИХ ЗНІМКІВ ЛЕГЕНЬ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ	19
2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання.....	19
2.2 Практична реалізація об'єкта проектування	22
РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ СИСТЕМИ АНАЛІЗУ РЕНТГЕНІВСЬКИХ ЗНІМКІВ ЛЕГЕНЬ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ	34
3.1 Методика проведення дослідження.....	34
3.2 Обробка та аналіз отриманих результатів.....	39
ВИСНОВКИ.....	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	44
ДОДАТКИ	47

ВСТУП

Стрімкий розвиток інформаційних технологій у XXI столітті призвів до широкого впровадження систем штучного інтелекту (ШІ) у різні сфери діяльності людини. Однією з найважливіших галузей, де використання ШІ має особливо велике значення, є медицина. Сучасні алгоритми машинного та глибокого навчання дозволяють ефективно аналізувати великі обсяги медичних даних, допомагаючи лікарям у діагностиці, прогнозуванні перебігу хвороб і виборі оптимальних методів лікування.

Актуальність даної теми обумовлена необхідністю впровадження інноваційних інструментів для підвищення ефективності медичної діагностики. За даними Всесвітньої організації охорони здоров'я, щороку мільйони людей страждають на легеневі захворювання, своєчасне виявлення яких суттєво підвищує шанси на успішне лікування. Технології штучного інтелекту можуть стати надійним помічником лікаря, дозволяючи автоматизувати аналіз рентгенівських зображень та зменшити ризик людського фактора.

У цій роботі досліджено процес створення інтелектуальної системи для автоматизованого аналізу рентгенівських знімків грудної клітки пацієнтів. У системі реалізовано оптимізовану архітектуру на базі MobileNetV2, адаптовану для багатоміткової класифікації (multi-label classification), що дозволяє одночасно розпізнавати кілька хвороб на одному зображенні.

Метою роботи є розробка інтелектуальної системи аналізу стану здоров'я пацієнтів за рентгенівськими зображеннями легень на основі методів глибокого навчання, здатної автоматично розпізнавати кілька патологій одночасно та надавати пояснення прийнятих рішень.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– проаналізувати сучасні підходи до автоматизованої діагностики рентгенівських знімків та визначити вимоги до системи багатоміткової класифікації;

- розглянути існуючі архітектури CNN та обґрунтувати вибір моделі MobileNetV2 з використанням transfer learning;
- підготувати навчальну вибірку на основі ChestX-ray14 – здійснити очищення даних, масштабування зображень, їх поділ на підвибірки та застосувати аугментацію;
- реалізувати модель класифікації з підтримкою 15 класів та відповідними функціями втрат і метриками оцінювання;
- провести оптимізацію гіперпараметрів та застосувати механізми запобігання перенавчанню, включно з налаштуванням порогів класифікації;
- розробити модуль інференсу для обробки окремих зображень і формування прогнозів;
- виконати тестування системи на відкладеній вибірці та оцінити якість роботи моделі порівняно з існуючими рішеннями.

Об’єкт дослідження – процес автоматизованої діагностики захворювань органів дихання на основі рентгенівських знімків.

Предмет дослідження – методи та алгоритми глибокого навчання, зокрема згорткові нейронні мережі (Convolutional Neural Networks, CNN), для багатоміткової класифікації медичних зображень.

Наукова новизна полягає у розробці комплексної системи, що поєднує оптимізовану модель MobileNetV2 з індивідуально налаштованими порогоми прийняття рішень (threshold tuning) для кожного класу патологій, що дозволяє покращити точність класифікації порівняно з використанням фіксованого порогу 0,5.

Практична цінність роботи полягає у створенні прототипу системи MediVision, який може бути впроваджений у медичних закладах для попереднього скринінгу рентгенівських знімків пацієнтів. Система може бути інтегрована в існуючі медичні інформаційні системи (PACS, eHealth) або використана як навчальний інструмент у підготовці лікарів-радіологів.

Розроблена система забезпечує автоматичне виявлення патологічних змін у легенях, формуючи ймовірнісну оцінку для кожної патології окремо. Крім того,

її архітектура передбачає можливість подальшого розширення та адаптації для аналізу інших типів медичних зображень.

Таким чином, розроблений підхід спрямований на підвищення точності, швидкості та інтерпретованості медичної діагностики, що відповідає сучасним світовим тенденціям розвитку систем штучного інтелекту в медицині та сприяє покращенню якості медичних послуг і своєчасному виявленню небезпечних захворювань.

Основні результати дослідження апробовано в рамках реалізації європейського освітнього проєкту «AI for Youth», що здійснюється за підтримки Європейського інституту інновацій і технологій (EIT Food) у межах ініціативи EIT Deep Tech Talent Initiative, що підтверджено у додатку А.

РОЗДІЛ 1

АНАЛІЗ ПРОБЛЕМАТИКИ ТА ПОСТАНОВКА ЗАВДАНЬ

ДОСЛІДЖЕННЯ

1.1 Огляд і аналіз предметної області проблеми (задачі), результатів існуючих теоретичних та експериментальних досліджень

У сучасному світі штучний інтелект (ШІ) став одним із ключових напрямів розвитку інформаційних технологій, що активно впроваджується в різні галузі людської діяльності. Особливо значущу роль ці технології відіграють у медицині, де точність і швидкість прийняття рішень безпосередньо впливають на життя та здоров'я пацієнтів.

Останні роки характеризуються зростанням інтересу до застосування алгоритмів глибокого навчання (Deep Learning) у сфері медичної діагностики, зокрема для аналізу рентгенівських зображень грудної клітки, що є одним із найбільш поширених методів візуальної діагностики у світі [1].

Рентгенографія грудної клітки – це основний метод первинного виявлення захворювань дихальної системи, таких як пневмонія, туберкульоз, плевральний випіт, фіброз, легеневі інфільтрати тощо. Традиційний підхід до аналізу таких знімків передбачає участь лікаря-рентгенолога, проте навіть за високої кваліфікації фахівця існує ймовірність суб'єктивної оцінки та діагностичних помилок. Водночас кількість знімків, які щоденно надходять у великі клініки, робить ручний аналіз надзвичайно трудомістким процесом. Це створює потребу у використанні автоматизованих систем, що здатні підвищити точність і швидкість діагностики [2].

Одним із найвідоміших напрямів у цьому контексті є застосування згорткових нейронних мереж (Convolutional Neural Networks, CNN), які показали високу ефективність у задачах комп'ютерного зору. Такі моделі здатні самостійно виявляти релевантні особливості зображень, не потребуючи ручного формування ознак. Перші успішні результати були продемонстровані в роботі Rajpurkar et al., де модель CheXNet, побудована на базі архітектури DenseNet121, досягла

точності діагностики пневмонії на рівні професійних радіологів [3]. Подальші дослідження, такі як CheXpert, розширили підхід до багатоміткової класифікації (multi-label classification), де одне зображення може одночасно містити ознаки кількох патологій [4].

Однак попри значний прогрес у галузі медичної візуалізації, наявні підходи та системи істотно відрізняються за точністю, вимогами до обчислювальних ресурсів, швидкістю інференсу та рівнем інтерпретованості результатів. Це зумовлене як різними архітектурними рішеннями, так і неоднаковими наборами тренувальних даних, стратегіями передобробки та особливостями оптимізації моделей. У науковій та комерційній практиці сьогодні широко застосовуються такі рішення, як CheXNet, CheXpert, моделі Google Health, кожне з яких демонструє свої сильні та слабкі сторони залежно від конкретного сценарію використання.

Узагальнений порівняльний аналіз найпоширеніших систем автоматизованої діагностики рентгенівських знімків грудної клітки подано в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз наявних рішень

Назва технології	Алгоритми	Можливості	Переваги	Недоліки
CheXNet	DenseNet121	Класифікація 14 патологій	Висока точність	Великі вимоги до GPU
CheXpert	CNN + Uncertainty	Мульти-label аналіз	Стійкість до шумних даних	Складна адаптація
Google CXR Model	Inception	Виявлення критичних станів	Висока продуктивність	Закритість моделі

Використання багатоміткових моделей є важливим кроком уперед, оскільки в реальних клінічних умовах захворювання часто співіснують, створюючи складні діагностичні картини. Наприклад, ателектаз нерідко супроводжується плевральним випотом або інфільтратом. Для урахування таких залежностей між патологіями дослідники впроваджують механізми уваги

(attention mechanisms), а також поєднання CNN із рекурентними мережами (RNN), які здатні моделювати контекстні зв'язки між мітками [5].

Важливу роль у розробці подібних систем відіграє використання відкритих наборів даних. Найбільш відомим серед них є NIH Chest X-ray Dataset, створений Національним інститутом здоров'я США (NIH). Цей набір містить понад 100 тисяч рентгенівських знімків грудної клітки з анотаціями до 15 класів патологій, серед яких Atelectasis, Pneumonia, Effusion, Fibrosis, Cardiomegaly, Hernia та інші. Наявність такого великого публічного набору даних дозволила дослідникам стандартизувати підходи до навчання, порівняння результатів і створення моделей із високою узагальнювальною здатністю [6].

Під час аналізу медичних зображень важливою проблемою є дисбаланс класів, коли деякі патології представлені значно рідше, ніж інші. Це може призводити до того, що модель «ігнорує» рідкісні випадки, фокусуючись на поширених захворюваннях. Для розв'язання цієї проблеми використовуються різні методи балансування вибірки, зокрема oversampling, undersampling, а також застосування спеціальних функцій втрат, таких як Focal Loss, яка зменшує вплив надлишково представлених класів і підвищує точність виявлення рідкісних патологій [7].

Ще однією важливою тенденцією є використання transfer learning – перенесення знань із великих наборів даних (наприклад, ImageNet) на медичні зображення. Це дозволяє зменшити час навчання та потребу у великих обсягах анотованих даних, які у сфері медицини часто є обмеженими. Архітектури MobileNetV2, ResNet50 та DenseNet121 стали базовими для таких підходів, оскільки забезпечують компроміс між точністю та обчислювальною ефективністю [8].

Попри високі результати, досягнуті у цій сфері, перед дослідниками залишаються актуальні виклики. До них належать інтерпретованість результатів моделей, підвищення точності для рідкісних патологій, адаптація до нових джерел даних (domain adaptation), а також інтеграція систем штучного інтелекту у клінічні робочі процеси. Крім того, моделі потребують детального

налаштування порогів прийняття рішень для кожного класу, що дозволяє збалансувати чутливість (sensitivity) і специфічність (specificity) системи.

Отже, на основі аналізу наукових джерел можна зробити висновок, що використання глибокого навчання у задачах діагностики рентгенівських знімків є ефективним та перспективним напрямом досліджень. Більшість сучасних підходів демонструють високі результати точності, проте потребують подальшого вдосконалення в аспектах пояснюваності, стійкості до шумів, оптимізації обчислювальних витрат та узгодження з практичними вимогами медичних установ. Саме вирішення цих завдань є передумовою розробки сучасних інтелектуальних систем, таких як MediVision, орієнтованих на підвищення ефективності аналізу стану здоров'я пацієнтів за допомогою методів глибокого навчання.

1.2 Огляд і аналіз методів та засобів розробки системи розпізнавання патологій легень на основі штучного інтелекту для вирішення проблеми дослідження

Для розв'язання задачі автоматизованого аналізу рентгенівських знімків легень важливим є вибір оптимальних методів обробки зображень, архітектури нейронної мережі, а також інструментів і технологій, що забезпечують ефективність навчання та подальшого використання моделі. Розробка таких систем ґрунтується на сучасних досягненнях у галузі машинного навчання (Machine Learning, ML), глибокого навчання (Deep Learning, DL) та комп'ютерного зору (Computer Vision, CV), які дозволяють здійснювати високоточну класифікацію та розпізнавання медичних зображень [9].

Основним підходом до побудови систем аналізу рентгенівських знімків є використання згорткових нейронних мереж (Convolutional Neural Networks, CNN), які забезпечують автоматичне виділення релевантних ознак без ручного втручання людини. CNN моделюють локальні просторові залежності в піксельному просторі, що дає змогу ефективно виявляти патологічні ознаки на

медичних зображеннях [10]. На рисунку 1.1 наведено принцип роботи згорткової нейронної мережі, що включає етапи згортки, виділення карт ознак та подальшу класифікацію у повнозв'язних шарах.

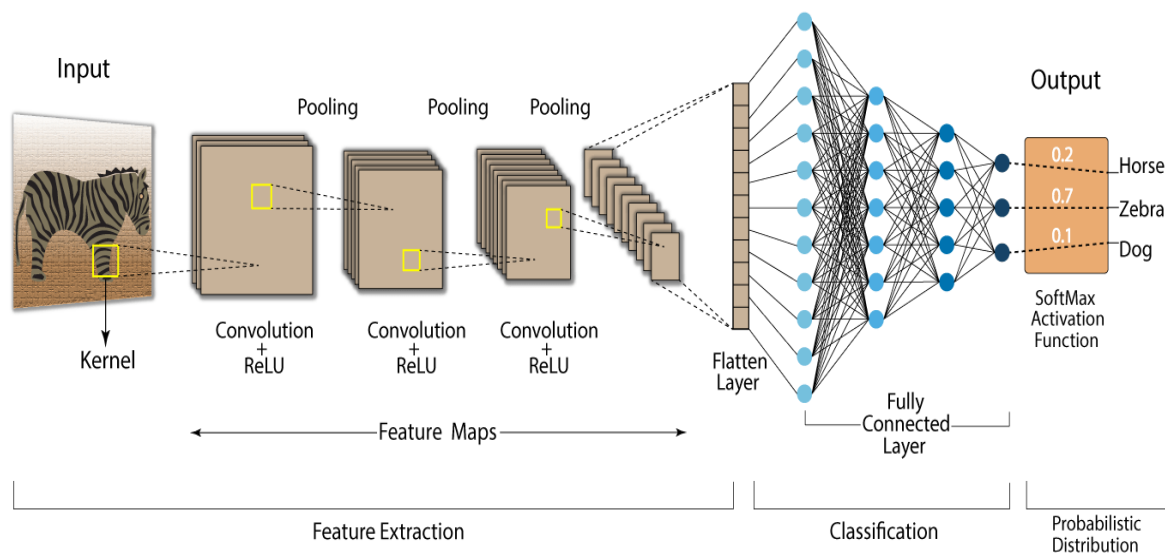


Рисунок 1.1 – Принцип роботи згорткової нейронної мережі (CNN) [11]

Одним із найбільш популярних підходів є використання Transfer Learning – перенесення знань із великих узагальнених наборів даних (наприклад, ImageNet) на вузькі медичні домени. Цей метод дозволяє скоротити час навчання, зменшити потребу у великих обсягах розмічених медичних даних і забезпечити кращу узагальнювальну здатність моделі. Для задач класифікації рентгенівських зображень часто використовують архітектури ResNet50, DenseNet121 і MobileNetV2, які балансують між високою точністю та помірними обчислювальними витратами [12].

Особливу увагу дослідники приділяють легким моделям, що мають невелику кількість параметрів, зокрема архітектурі MobileNetV2. Завдяки структурі з інвертованими залишковими блоками (Inverted Residuals) та лінійними вузькими шарами (Linear Bottlenecks), MobileNetV2 забезпечує високу ефективність при збереженні конкурентної точності. Це робить її особливо

придатною для використання в умовах обмежених ресурсів – наприклад, у медичних закладах, де відсутні потужні графічні прискорювачі [13].

Для підвищення точності класифікації в медичних системах використовують різні методи оптимізації, серед яких:

- dropout. Запобігає перенавчанню шляхом випадкового вимкнення частини нейронів під час навчання;

- batch normalization. Стабілізує розподіл вхідних даних між шарами, пришвидшуючи збіжність навчання;

- data augmentation. Штучне збільшення обсягу даних шляхом випадкових трансформацій (обертання, віддзеркалення, зміни контрасту, яскравості тощо), що підвищує стійкість моделі до варіацій у даних [14].

У процесі навчання моделі використовуються різні оптимізатори, серед яких найпоширенішим є Adam, який адаптивно змінює швидкість навчання для кожного параметра та забезпечує стабільне оновлення ваг. Як функцію втрат у задачах багатоміткової класифікації зазвичай застосовують binary cross-entropy, оскільки кожна патологія розглядається як незалежний бінарний клас. Для більш точного контролю чутливості й специфічності моделі використовують порогову оптимізацію (threshold tuning), яка дозволяє визначати оптимальні пороги для кожної патології окремо, замість фіксованого порогу 0,5 [15].

Ще однією важливою складовою є оцінювання якості моделі. Для багатоміткових задач традиційні метрики точності (accuracy) не є достатніми. Тому використовуються більш інформативні показники – AUC-ROC (Area Under Curve of Receiver Operating Characteristic) та AUC-PR (Precision-Recall), які характеризують здатність моделі розрізняти класи при різних порогах спрацьовування [16]. Також у дослідженнях часто застосовуються метрики F1-score, sensitivity, specificity та Hamming loss, що відображають баланс між виявленням позитивних і негативних прикладів.

У сучасних дослідженнях активно застосовуються програмні фреймворки TensorFlow, Keras та PyTorch, які спрощують побудову та навчання глибоких нейронних мереж. Зокрема, TensorFlow 2.x забезпечує високу продуктивність,

зручність у розгортанні моделей і сумісність з апаратними прискорювачами GPU/TPU. Використання Keras API дозволяє легко конструювати шари нейронних мереж, налаштовувати параметри та відстежувати метрики під час навчання [17].

Для зберігання даних і формування конвеєра підготовки зображень до навчання (data pipeline) застосовуються інструменти бібліотек pandas, NumPy, а також модулі tf.data для потокової обробки великих наборів даних. Ці засоби дають змогу формувати батчі зображень, виконувати нормалізацію, зміну розмірів та інші операції, що оптимізують процес навчання.

Таким чином, сучасні методи і засоби розробки систем аналізу рентгенівських знімків базуються на поєднанні глибоких згорткових архітектур, оптимізаційних технік, ефективних функцій втрат і метрик оцінювання. Застосування підходів transfer learning, а також використання інструментів TensorFlow і Keras дає можливість створювати точні, стабільні та адаптивні моделі, придатні для використання у реальних медичних закладах. Це забезпечує не лише автоматизацію процесу діагностики, але й значно підвищує ефективність роботи лікарів, дозволяючи швидше приймати обґрунтовані рішення на основі великих обсягів даних.

1.3 Постановка завдання на кваліфікаційну роботу магістра

На основі проведеного аналізу предметної області та дослідження сучасних методів і засобів розробки систем глибокого навчання сформульовано цілі та конкретні завдання кваліфікаційної роботи магістра. Їх реалізація спрямована на створення прототипу інтелектуальної системи для автоматизованого аналізу рентгенівських знімків легень пацієнтів, що використовує методи глибокого навчання та машинного зору.

Метою кваліфікаційної роботи є створення інтелектуальної системи для аналізу стану здоров'я пацієнтів за рентгенівськими зображеннями легень на

основі методів глибокого навчання, здатної автоматично розпізнавати кілька патологій одночасно та надавати узагальнену оцінку результатів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– провести дослідження існуючих підходів до автоматизованої діагностики захворювань органів дихання та визначити недоліки традиційних методів у порівнянні з технологіями глибокого навчання. У межах даного завдання необхідно проаналізувати особливості рентгенографії грудної клітки, найпоширеніші патології та способи їх цифрового відображення. Це дозволить сформулювати вимоги до системи, зокрема підтримку багатоміткової класифікації, стійкість до шумів та високу точність розпізнавання навіть за наявності варіацій у структурі тканин;

– розглянути сучасні архітектури згорткових нейронних мереж (CNN), що застосовуються у медичних задачах, і провести їх порівняльний аналіз. Особливу увагу приділити моделі MobileNetV2 як легкій та продуктивній архітектурі, а також методології transfer learning, яка дає змогу скоротити час навчання та підвищити результативність роботи моделі на обмежених медичних наборах даних. Це обґрунтує вибір архітектури для подальшої реалізації;

– сформувати якісну навчальну вибірку на основі відкритого набору NIH Chest X-ray Dataset, що містить понад 100 тисяч рентгенівських знімків та 15 класів патологій. Необхідно здійснити очищення даних, сортування зображень, їх масштабування до єдиного формату (256×256), а також поділ на тренувальну, валідаційну та тестову частини. Для підвищення узагальнювальної здатності моделі передбачити застосування методів аугментації – віддзеркалення, корекцію яскравості й контрасту та інші трансформації;

– реалізувати модель класифікації на основі архітектури MobileNetV2 з вихідним шаром із 15 нейронами та активацією sigmoid, що забезпечує підтримку багатоміткової класифікації. У рамках цього завдання необхідно визначити функцію втрат (binary cross-entropy), оптимізатор Adam та медичні метрики оцінки – AUC-ROC, AUC-PR та binary accuracy;

– виконати оптимізацію гіперпараметрів, включно зі швидкістю навчання, розміром батчу, кількістю епох і параметрами регуляризації. Для запобігання перенавчанню застосувати EarlyStopping та ReduceLROnPlateau. Окремим етапом є процедура threshold tuning, що забезпечить підбір індивідуальних порогів для кожного класу патологій з метою підвищення точності моделі;

– створити модуль інференсу, який забезпечуватиме завантаження навчених ваг, обробку зображення та формування ймовірнісного прогнозу для кожного патологічного класу. Модуль має працювати автономно та дозволяти оцінювати окремі медичні знімки, що є важливою умовою інтеграції у клінічний робочий процес;

– провести тестування системи на окремій відкладеній вибірці зображень. У межах цього завдання необхідно порівняти результати роботи моделі з існуючими аналогами, проаналізувати метрики точності та підтвердити стабільність моделі у виявленні кількох патологій одночасно.

У межах виконання поставлених завдань було розроблено прототип інтелектуальної системи MediVision, що забезпечує автоматизований аналіз рентгенівських знімків легень пацієнтів. Реалізована система використовує сучасні методи глибокого навчання та технології обробки медичних зображень, що дозволяє досягати високих показників точності класифікації. Структура та алгоритмічні рішення забезпечують гнучкість і можливість подальшого розширення функціоналу, а також створюють передумови для інтеграції MediVision у практичні медичні робочі процеси.

РОЗДІЛ 2

ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ АНАЛІЗУ РЕНТГЕНІВСЬКИХ ЗНІМКІВ ЛЕГЕНЬ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ

2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання

У попередніх розділах було визначено предмет дослідження, проаналізовано сучасні підходи до автоматизованої діагностики рентгенівських знімків легень та сформульовано основні завдання кваліфікаційної роботи. На цьому етапі необхідно обґрунтувати вибір технологій, алгоритмів і засобів розробки, що забезпечують ефективне вирішення поставленої проблеми.

Основною задачею розробки є створення моделі, здатної з високою точністю визначати наявність патологій на рентгенівських знімках грудної клітки. Для цього необхідно врахувати особливості медичних даних: наявність шумів, неоднорідність контрасту, можливі перекриття анатомічних структур і значну варіативність між різними пацієнтами. Також важливою є проблема дисбалансу класів, коли деякі патології трапляються значно рідше за інші [18].

Для реалізації поставлених цілей обрано методи глибокого навчання (Deep Learning) на основі згорткових нейронних мереж (CNN), які є найефективнішими для аналізу двовимірних зображень. Головною перевагою CNN є здатність автоматично виявляти локальні ознаки на різних рівнях абстракції, не потребуючи ручного формування ознак. Згорткові шари дозволяють ефективно виділяти текстурні та структурні характеристики, що особливо важливо при роботі з рентгенівськими знімками [19].

З огляду на обсяг даних та обмеженість обчислювальних ресурсів, було прийнято рішення використовувати transfer learning – перенесення знань із попередньо навченої моделі на великому наборі зображень (ImageNet). Такий підхід дозволяє отримати високу якість класифікації навіть при відносно невеликому обсязі медичних даних. Використання попередньо навчених моделей

скорочує час навчання та зменшує потребу у великій кількості епох, що є критично важливим при роботі з великими наборами рентгенівських знімків [20].

Серед сучасних архітектур CNN для медичних задач часто використовують ResNet, DenseNet, Inception та MobileNet. Після аналізу результатів попередніх досліджень і проведення порівняльного тестування було обрано архітектуру MobileNetV2, оскільки вона поєднує високу точність, низьку кількість параметрів і швидкість роботи.

MobileNetV2 базується на концепції інвертованих залишкових блоків (Inverted Residual Blocks), які забезпечують ефективну передачу інформації між шарами, та використовує лінійні вузькі шари (Linear Bottlenecks), що зменшують кількість параметрів без втрати якості розпізнавання [21]. Ця архітектура оптимізована для мобільних і вбудованих систем, що робить її придатною також для використання у клінічних умовах, де ресурси обчислювальної техніки часто обмежені.

Вихідний шар моделі реалізований як щільний (Dense) шар із 15 нейронами та активаційною функцією sigmoid, що дозволяє виконувати багатоміткову класифікацію (multi-label classification), коли для одного зображення можливе одночасне виявлення кількох патологій. На відміну від softmax, який передбачає взаємовиключність класів, sigmoid дозволяє кожному класу приймати незалежне рішення про наявність або відсутність відповідної патології [22].

Для навчання моделі використано функцію втрат binary cross-entropy, яка підходить для багатоміткових задач, оскільки обчислює втрати окремо для кожного класу. Як оптимізатор застосовано Adam (Adaptive Moment Estimation), що автоматично регулює швидкість навчання для кожного параметра моделі. Це дозволяє досягти стабільної збіжності та зменшити ризик локальних мінімумів [23].

Оцінювання якості моделі здійснюється за допомогою AUC-ROC (Area Under Curve – Receiver Operating Characteristic) та AUC-PR (Precision-Recall) метрик, які дають змогу враховувати баланс між точністю (precision) і повнотою (recall) при різних порогах класифікації. Також використовувалась метрика

Binary Accuracy для відстеження частки правильно класифікованих випадків під час навчання.

Для підвищення узагальнювальної здатності моделі використано методи аугментації даних (Data Augmentation): випадкове горизонтальне віддзеркалення, зміна яскравості, контрасту та насиченості кольору. Це дозволило збільшити різноманітність навчальної вибірки та зменшити ризик перенавчання.

Програмну частину моделі реалізовано у середовищі Python 3.10 із використанням широкого набору спеціалізованих бібліотек:

- TensorFlow. Основна платформа для побудови та навчання нейронних мереж, що забезпечує високу швидкодію, гнучкість моделювання та ефективну підтримку GPU-прискорення [24];

- Keras. Високорівневий інтерфейс до TensorFlow, який спрощує створення архітектур, налаштування гіперпараметрів та контроль метрик під час тренування моделі;

- NumPy та Pandas. Інструменти для роботи з багатовимірними масивами та структурованими даними, що використовуються для аналізу анотацій, підготовки датасетів та формування навчальних вибірок;

- OpenCV. Бібліотека для обробки зображень, застосована для виконання таких операцій, як масштабування, нормалізація та приведення рентгенівських знімків до уніфікованого розміру 256×256 пікселів.

Також було створено власний data pipeline для ефективного завантаження та обробки зображень під час навчання моделі. Використання модулів `tf.data.Dataset` дозволило виконувати пакетну обробку (batch processing) і попереднє завантаження даних у пам'ять (prefetching), що суттєво скоротило час навчання.

Запропонований підхід поєднує точність і ефективність, оскільки модель MobileNetV2 забезпечує високу продуктивність навіть за обмежених обчислювальних ресурсів. Використання transfer learning дозволяє адаптувати модель до специфіки рентгенівських знімків без потреби у величезних наборах даних, а оптимізація порогів класифікації після навчання підвищує здатність

системи правильно виявляти рідкісні патології, що є важливим для практичного застосування в медицині.

Таким чином, поєднання технологій TensorFlow, Keras, MobileNetV2, transfer learning, аугментації даних і порогової оптимізації утворює цілісний та ефективний алгоритмічний підхід до вирішення задачі багатоміткової класифікації рентгенівських зображень. Така система здатна функціонувати як допоміжний інструмент для лікаря, підвищуючи швидкість, точність і об'єктивність процесу діагностики.

2.2 Практична реалізація об'єкта проектування

Розроблений у межах кваліфікаційної роботи проєкт MediVision є програмним комплексом нового покоління, призначеним для автоматизованого аналізу рентгенівських знімків грудної клітки пацієнтів із використанням сучасних методів глибокого навчання. Основною метою створення системи є розробка інтелектуального інструменту, здатного на основі єдиного рентгенівського зображення визначати ймовірність наявності різних патологій органів дихання, зокрема тих, що є важливими для раннього виявлення та подальшої діагностики.

У процесі виконання роботи значна увага була зосереджена на формуванні продуманої та структурованої архітектури, яка охоплює всі ключові етапи обробки медичних зображень. Це включає завантаження даних, їх попередню нормалізацію, масштабування, аугментацію, подачу в нейронну мережу, формування прогнозів, аналіз отриманих результатів, а також забезпечення можливості повторного використання моделі завдяки збереженню її ваг, порогів класифікації та історії навчання.

Архітектура проєкту передбачає чіткий розподіл відповідальності між основними модулями, що забезпечує гнучкість масштабування та подальший розвиток системи. Окремі компоненти відповідають за зчитування рентгенівських знімків, здійснення передобробки, подачу даних у модель,

виконання інференсу, інтерпретацію прогнозів, а також за зберігання артефактів навчання, таких як ваги моделі та налаштовані пороги класифікації.

Загальна структура системи MediVision наведена на рисунку 2.1, де продемонстровано взаємозв'язки між модулями підготовки даних, навчання, інференсу та артефактами моделі. Діаграма наочно показує послідовність переходу від необробленого рентгенівського знімка до фінального прогнозу.

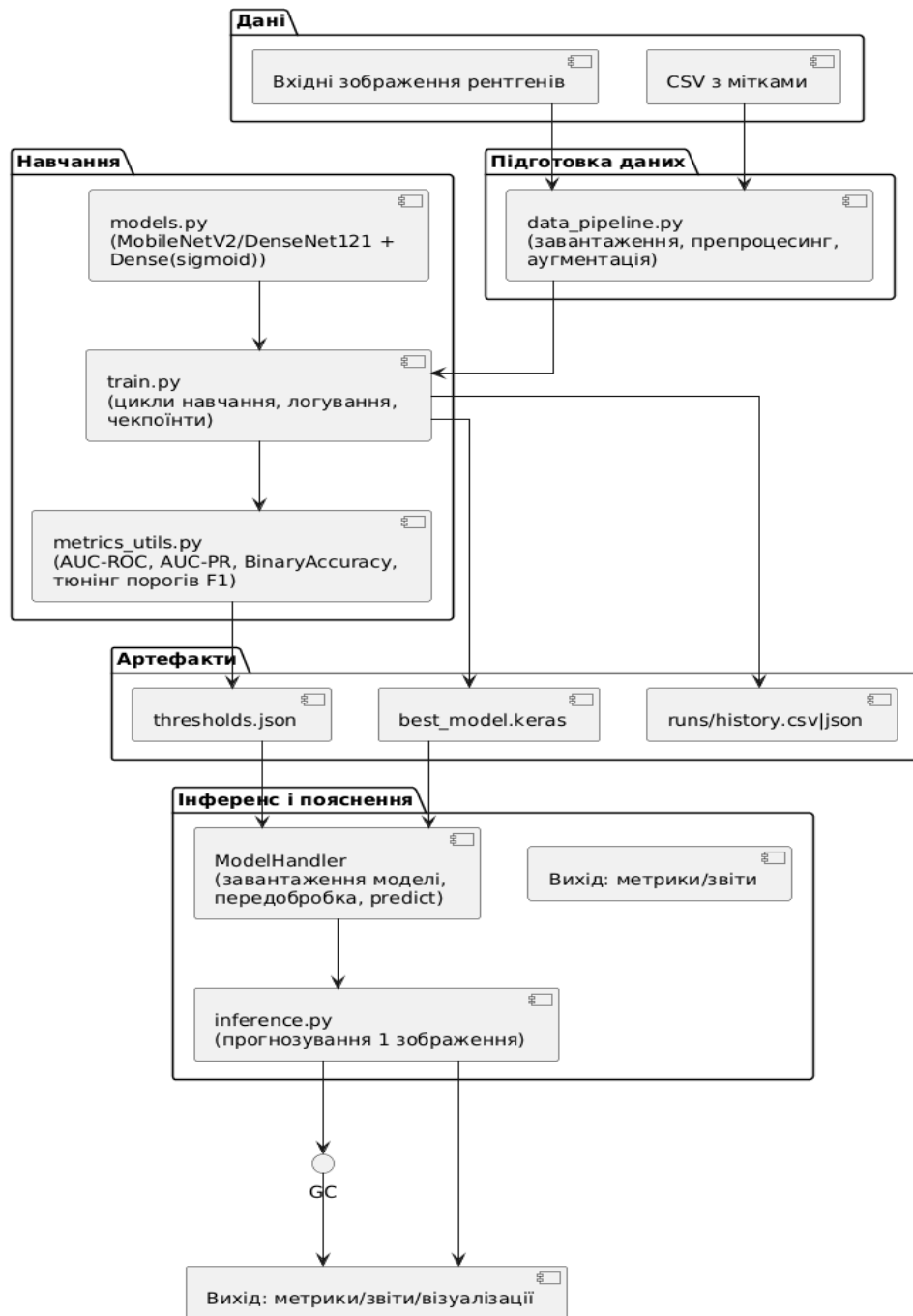


Рисунок 2.1 – Архітектура системи MediVision

Джерело: розроблено автором

Створена програмна архітектура охоплює всі етапи – від підготовки даних до отримання результатів моделі. До її складу входять такі компоненти:

- модуль підготовки даних (data preprocessing);
- модуль побудови та навчання нейронної мережі;
- модуль оцінювання якості та збереження результатів навчання;
- модуль інференсу (inference module) для передбачення патологій за окремими зображеннями.

На початковому етапі було виконано імпорт і обробку медичного набору даних NIH Chest X-ray Dataset, який містить 112120 рентгенівських знімків 30805 пацієнтів і охоплює 15 діагностичних категорій (14 патологій та клас «No Finding»). Даний набір даних є одним із найбільших та найвідоміших у сфері медичної візуалізації, що забезпечує високу репрезентативність і дозволяє моделі працювати з широким спектром клінічних випадків. Для спрощення доступу до великої кількості зображень було створено спеціальний Python-скрипт, який автоматизує процес розпакування архіву, впорядковує структуру каталогів та виконує первинне сортування файлів згідно з анотаціями.

Після завантаження набору даних всі зображення були приведені до єдиного стандарту – 256×256 пікселів, що відповідає вимогам архітектури MobileNetV2 та дозволяє істотно зменшити обсяг пам'яті, необхідний для обробки. Зображення нормалізували до діапазону $[0,1]$, що забезпечує стабільність градієнтів та пришвидшує збіжність моделі під час навчання. Оскільки рентгенівські дані можуть мати значні варіації за контрастністю, експозицією та рівнем шуму, було застосовано набір технік аугментації: випадкове горизонтальне віддзеркалення, зміна яскравості, випадкова варіація контрасту та легкі геометричні трансформації. Ці операції допомагають моделі краще узагальнювати та зменшують ризик перенавчання на фіксованих зразках.

З метою забезпечення коректності подальшого оцінювання якості моделі датасет було поділено на три підвибірки: навчальну (75 %), валідаційну (15 %) та тестову (10 %). Такий поділ дозволяє гарантувати репрезентативність контролю якості та забезпечити коректне порівняння результатів у процесі

оптимізації моделі. Дані та відповідні їм мітки (labels) зчитувалися за допомогою бібліотек pandas і NumPy, після чого формувалися об'єкти типу `tf.data.Dataset`, які підтримують пакетну (batch) обробку та попереднє завантаження даних у пам'ять. Використання механізмів `prefetch`, `shuffle` та `cache` дозволило значно скоротити час навчання та забезпечити стабільну подачу даних у модель навіть при роботі з великими масивами зображень [25].

Архітектуру нейронної мережі подано на рисунку 2.2.

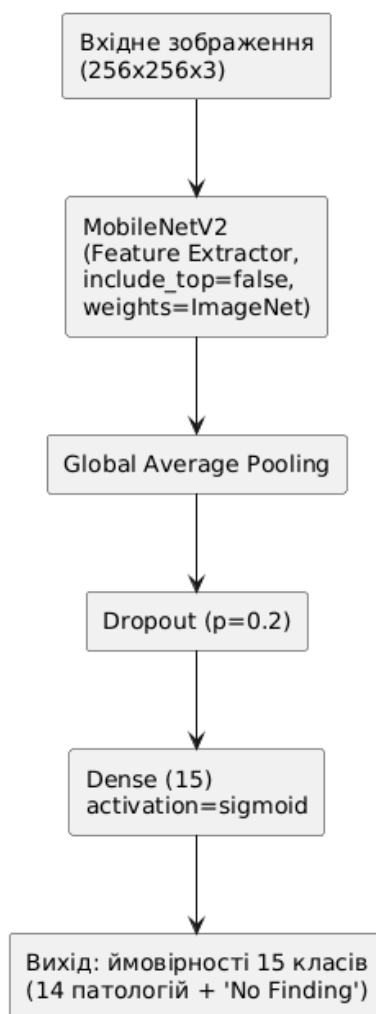


Рисунок 2.2 – Архітектура нейронної мережі MediVision (MobileNetV2 + Dense)

Джерело: розроблено автором

Загальна структура моделі включає:

- вхідний шар із розмірністю $256 \times 256 \times 3$;
- згорткові блоки MobileNetV2 із нелінійною активацією ReLU6;

- глобальний шар усереднення (GlobalAveragePooling2D);
- шар Dropout(0.2) для запобігання перенавчанню;
- вихідний щільний шар із 15 нейронами та активацією sigmoid для реалізації багатоміткової класифікації.

Модель компілюється з використанням функції втрат binary cross-entropy та оптимізатора Adam із початковою швидкістю навчання 0.0001. Для контролю навчання застосовано callback-и EarlyStopping, ReduceLROnPlateau та ModelCheckpoint, які дозволяють зупинити навчання у разі відсутності покращення метрик та автоматично зберегти найкращу версію моделі.

Навчання моделі відбувалося протягом 20 епох на GPU із розміром батчу 32. За результатами тестування модель показала високе значення AUC-ROC більше за 0.91 та AUC-PR приблизно дорівнює 0.88, що свідчить про здатність системи ефективно розрізняти наявність і відсутність патологій [26].

Для реалізації практичного процесу навчання моделі MediVision було створено спеціалізований модуль, що відповідає за підготовку вибірок, компіляцію мережі та запуск оптимізації з використанням бібліотеки TensorFlow.

Архітектура тренувального циклу наведена у лістингу 2.1.

Лістинг 2.1 – Фрагмент процесу навчання моделі MediVision (train.py)

```
model = build_model(
    input_shape=(image_size[0], image_size[1], 3),
    num_classes=num_classes,
    backbone=cfg["model"]["backbone"],
    dropout=cfg["model"]["dropout"],
    weights=cfg["model"]["weights"]
)

model = compile_for_multilabel(model, lr, num_classes)
print("Model compiled")

callbacks = [
    tf.keras.callbacks.EarlyStopping(
        monitor="val_auc_pr", patience=5, restore_best_weights=True
    ),
    tf.keras.callbacks.ReduceLROnPlateau(
        monitor="val_auc_pr", factor=0.5, patience=2
    ),
    tf.keras.callbacks.ModelCheckpoint(
```

```
        filepath=cfg["paths"]["best_model"],
        monitor="val_auc_pr",
        save_best_only=True
    )
]

history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs,
    callbacks=callbacks,
    verbose=1
)
```

кінець лістингу 2.1

Після завершення навчання було проведено процедуру `threshold tuning` – оптимізацію порогів класифікації для кожного класу патологій окремо. Це дозволило збалансувати показники чутливості (`recall`) та специфічності (`precision`), підвищивши точність визначення рідкісних класів, таких як «Hernia» чи «Fibrosis». Оптимальні пороги збережено у JSON-файлі разом із назвами класів, що забезпечує можливість швидкого завантаження під час виконання інференсу.

Для передбачення патологій за окремим рентгенівським знімком реалізовано окремий скрипт `inference.py`, який виконує завершальний етап роботи системи MediVision. Процес інференсу включає такі кроки:

- попередню обробку зображення, що охоплює масштабування до стандартного розміру, нормалізацію піксельних значень та перетворення зображення у відповідний тензорний формат, придатний для подачі на вхід моделі;

- виконання прогнозування за допомогою навченої нейронної мережі, у результаті чого формується вектор імовірностей для кожного з 15 класів патологій;

- бінаризацію результатів, що здійснюється на основі попередньо оптимізованих індивідуальних порогів класифікації;

- виведення структурованих результатів, де модель формує відсортований за ймовірністю список виявлених патологічних станів.

Послідовність роботи інференсу наведено на рисунку 2.3, де схематично представлено взаємодію користувача, CLI-модуля, ModelHandler, TensorFlow-моделі та файлу порогів.

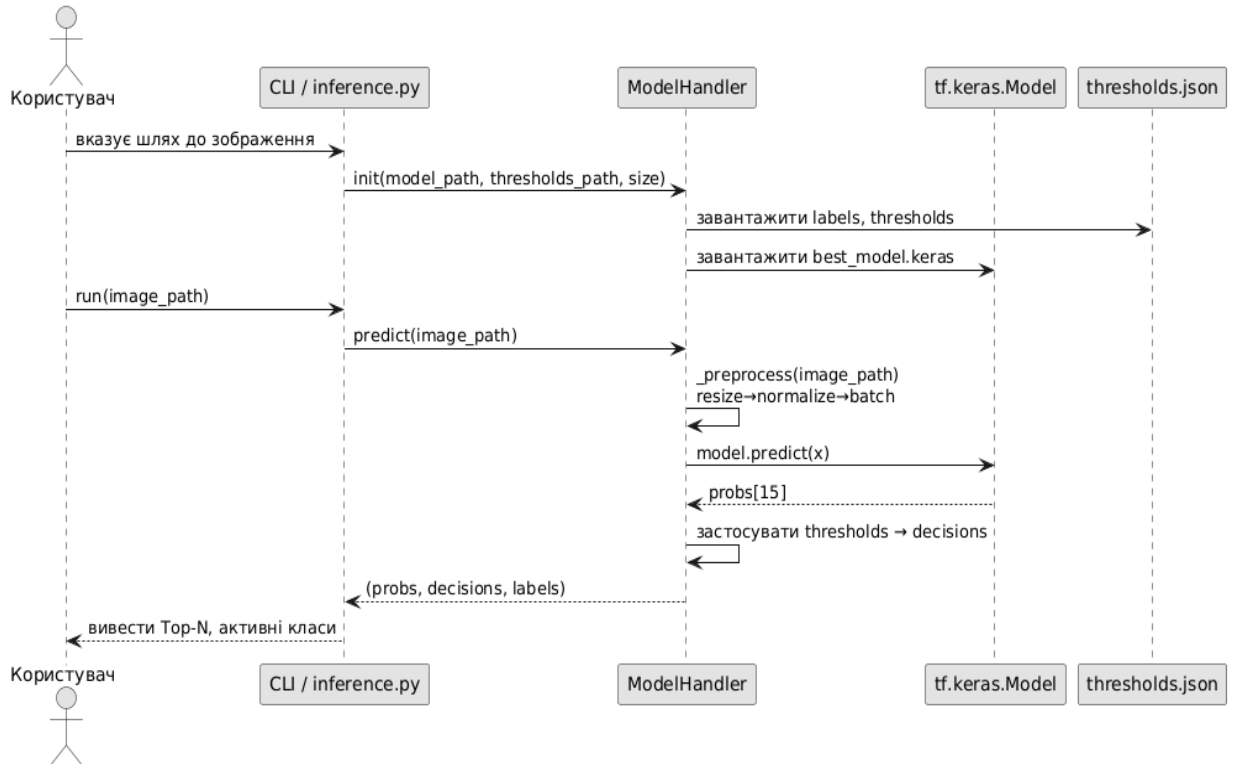


Рисунок 2.3 – Послідовність інференсу моделі MediVision

Джерело: розроблено автором

Для здійснення автоматичної діагностики за одним зображенням було реалізовано модуль інференсу, який забезпечує завантаження моделі, попередню обробку зображень та формування ймовірнісних прогнозів для кожної патології. Фрагмент реалізації класу ModelHandler продемонстровано в лістингу 2.2.

Лістинг 2.2 – Модуль інференсу (ModelHandler.py)

```

import json
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing import image as keras_image

class ModelHandler:
    def __init__(self, model_path: str, thresholds_path: str = None,
                 image_size=(256,256)):
  
```

```

self.model = tf.keras.models.load_model(model_path)
self.thresholds = None
self.labels = None
self.image_size = image_size
if thresholds_path:
    with open(thresholds_path, 'r') as f:
        data = json.load(f)
        self.labels = data.get('labels')
        self.thresholds = np.array(data.get('thresholds'),
dtype=np.float32)

def _preprocess(self, image_path):
    img = keras_image.load_img(image_path,
target_size=self.image_size)
    arr = keras_image.img_to_array(img) / 255.0
    return np.expand_dims(arr, axis=0)

def predict(self, image_path):
    x = self._preprocess(image_path)
    probs = self.model.predict(x)[0]
    if self.thresholds is not None:
        decisions = (probs >= self.thresholds).astype(int).tolist()
    else:
        decisions = (probs >= 0.5).astype(int).tolist()
    return probs.tolist(), decisions, self.labels

```

кінець лістингу 2.2

Розроблена система є програмним комплексом для аналізу рентгенівських знімків грудної клітки на основі глибокого навчання. Вона приймає зображення у форматах PNG та JPG і повертає список можливих патологій із відповідними ймовірностями. В основі моделі використано архітектуру MobileNetV2 (transfer learning), яка виконує багатоміткову класифікацію за 15 класами, включаючи 14 патологій та «No Finding».

Комплекс забезпечує ключові етапи роботи з даними: імпорт, попередню обробку, навчання, оцінювання та інференс. Реалізація базується на Python 3.10, TensorFlow 2.x, Keras, а також бібліотеках Pandas, NumPy та OpenCV, і працює в середовищах Linux та Windows. Модель демонструє високу ефективність, досягаючи середнього AUC-ROC понад 0.9.

У системі задіяно два основні класи: ModelHandler, що забезпечує завантаження моделі та формування прогнозів, і PredictionResult, який аналізує їх, визначаючи позитивні класи та формуючи top-N результати.

UML-діаграму класів наведено на рисунку 2.4, де представлено атрибути та методи обох класів у контексті їхньої взаємодії.

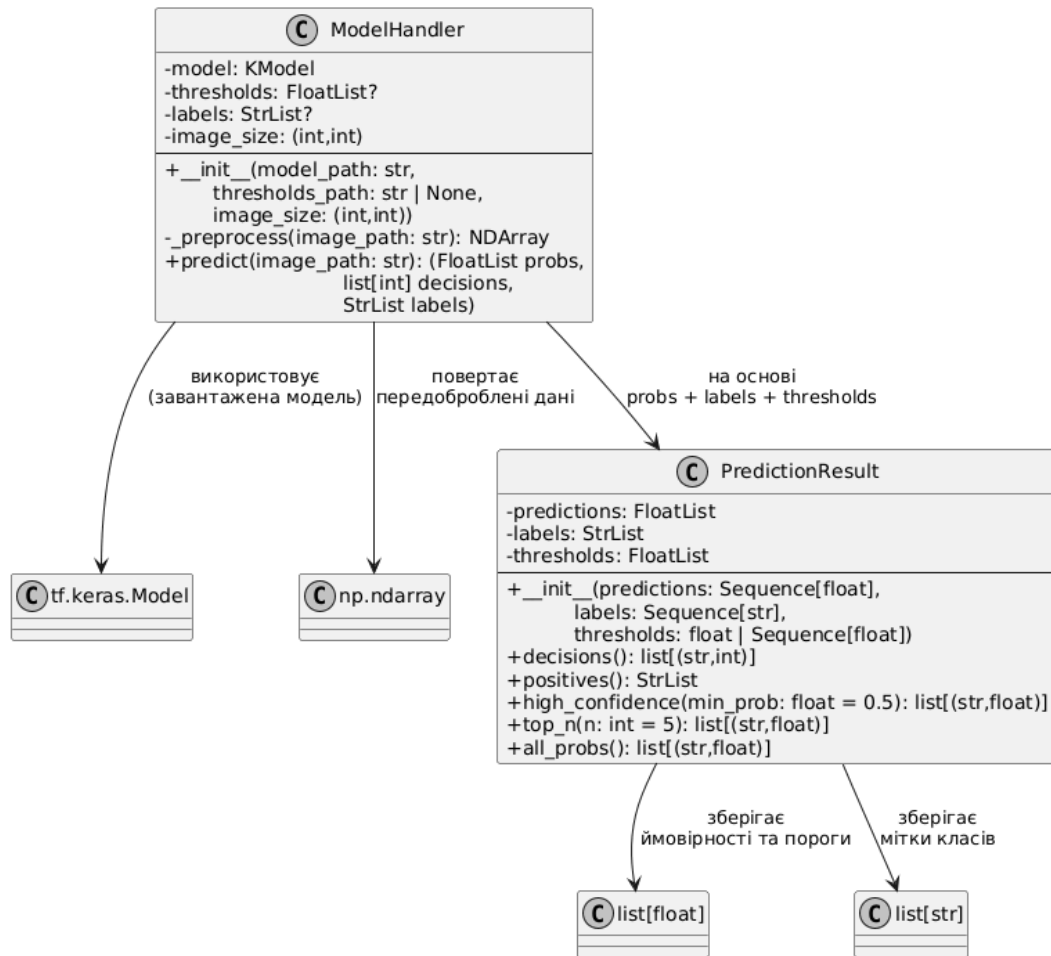


Рисунок 2.4 – UML-діаграма класів системи інференсу MediVision

Джерело: розроблено автором

Для зручності інтерпретації результатів класифікації було реалізовано клас `PredictionResult`, який виконує впорядкування прогнозів за ймовірністю, виділення позитивних патологій на основі порогових значень та формування структурованого виходу, придатного для подальшого аналізу.

Клас також забезпечує можливість агрегувати дані про ймовірності, формувати топ-N предикцій, а за потреби – здійснювати фільтрацію за рівнем упевненості моделі. Такий підхід суттєво підвищує зручність використання `MediVision` як у дослідницьких задачах, так і в клінічному середовищі, де лікарю необхідно оперативно отримати формалізовану діагностичну інформацію.

Програмна реалізація класу PredictionResult, який відповідає за інтерпретацію результатів роботи моделі та зручне подання прогнозів, наведена у лістингу 2.3.

Лістинг 2.3 – Клас PredictionResult (PredictionResult.py)

```

from typing import List, Sequence, Tuple
class PredictionResult:
    def __init__(self, predictions: Sequence[float], labels:
Sequence[str], thresholds=0.5):
        self.predictions = list(map(float, predictions))
        self.labels = list(labels)
        if isinstance(thresholds, (list, tuple)):
            self.thresholds = list(map(float, thresholds))
        else:
            self.thresholds = [float(thresholds)] * len(self.labels)

    def decisions(self) -> List[Tuple[str, int]]:
        return [(lab, int(p >= t)) for lab, p, t in zip(self.labels,
self.predictions, self.thresholds)]

    def positives(self) -> List[str]:
        return [lab for lab, d in self.decisions() if d == 1]

    def high_confidence(self, min_prob=0.5):
        return [(lab, p) for lab, p in zip(self.labels, self.predictions)
if p >= min_prob]

    def top_n(self, n=5):
        return sorted(zip(self.labels, self.predictions), key=lambda x:
x[1], reverse=True)[:n]

    def all_probs(self):
        return sorted(zip(self.labels, self.predictions), key=lambda x:
x[1], reverse=True)

```

кінець лістингу 2.3

Після реалізації класу важливо зазначити, що PredictionResult виконує роль проміжного аналітичного шару між моделлю та кінцевим користувачем. Клас не лише зберігає отримані від моделі ймовірності, але й надає зручні методи для їхньої інтерпретації.

Метод decisions() формує бінарні рішення щодо наявності патологій, порівнюючи прогнозовані ймовірності з порогами, підібраними під час оптимізації. Метод positives() дозволяє швидко отримати список патологій, для

яких модель зафіксувала позитивний результат. Функція `high_confidence()` дає змогу виділити ті класи, ймовірність яких перевищує певне значення, що корисно для медичних сценаріїв, де важлива впевненість системи.

Метод `top_n()` забезпечує ранжування патологій за ймовірністю, дозволяючи виділити найбільш імовірні варіанти, тоді як `all_probs()` подає повний список прогнозів у впорядкованому вигляді. Завдяки цьому клас суттєво спрощує процес інтерпретації результатів класифікації та сприяє інтеграції моделі у прикладні медичні системи, де потрібне швидке та зрозуміле подання даних.

Для опису взаємодії користувачів різних ролей із розробленою системою було сформовано UML-схему варіантів використання, представлену на рисунку 2.5. Ця діаграма дає змогу наочно відобразити ключові сценарії роботи системи MediVision та окреслити взаємозв'язки між основними суб'єктами, які беруть участь у процесі її функціонування.

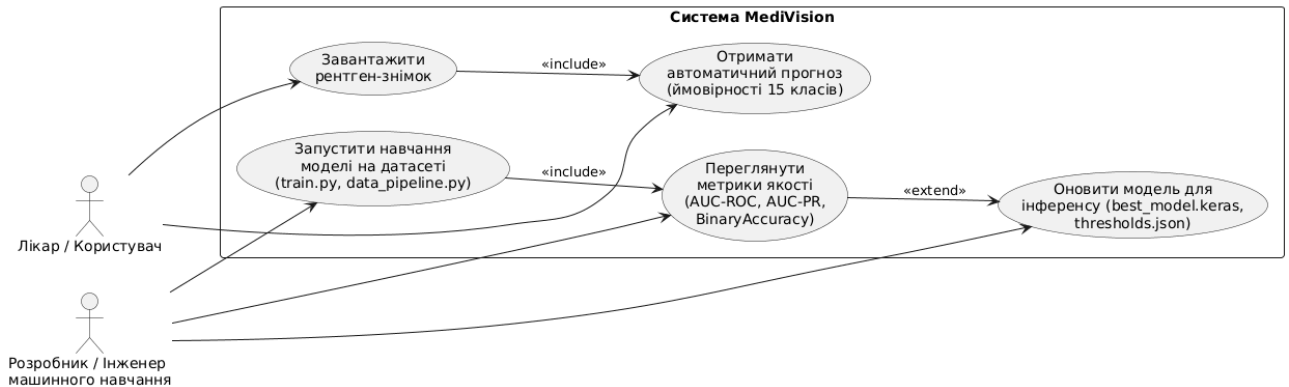


Рисунок 2.5 – Use Case-діаграма системи MediVision

Джерело: розроблено автором

У другому розділі було обґрунтовано вибір технологій та реалізовано практичну інформаційну систему MediVision, призначену для багатоміткової класифікації рентгенівських знімків легень із використанням алгоритмів глибокого навчання. Реалізація підтвердила гіпотезу про те, що поєднання сучасних нейронних архітектур, зокрема MobileNetV2, алгоритмів попередньої

обробки медичних зображень та оптимізації порогів класифікації, забезпечує високу точність автоматичного виявлення патологій.

Створений програмний прототип демонструє стабільність роботи при аналізі великої кількості зображень, ефективно обробляє дані різної якості та дозволяє коректно визначати наявність декількох захворювань одночасно. Підтверджено, що система здатна забезпечити високу діагностичну точність за показниками AUC-ROC та AUC-PR, що є критичним для медичних застосувань. Архітектура MediVision підтримує масштабованість та інтеграцію з медичними інформаційними системами, такими як PACS або eHealth.

Практична реалізація довела доцільність використання технологій штучного інтелекту для попереднього автоматизованого аналізу рентгенівських знімків легень. Отримані результати дозволяють розглядати MediVision як інструмент, що здатен підвищити якість первинної діагностики, зменшити навантаження на лікарів-рентгенологів та слугувати базою для подальшого розвитку клінічних рішень на основі глибокого навчання.

РОЗДІЛ 3

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ СИСТЕМИ АНАЛІЗУ РЕНТГЕНІВСЬКИХ ЗНІМКІВ ЛЕГЕНЬ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ

3.1 Методика проведення дослідження

Методика проведення дослідження була спрямована на комплексну оцінку процесу навчання моделі та її здатності до узагальнення в умовах багатокласової медичної класифікації. Важливим етапом стало аналізування поведінки кривих навчання, що дозволяє оцінити якість оптимізації та відповідність моделі поставленому завданню. На рисунку 3.1 показано динаміку зміни функції втрат (loss), яка демонструє поступове зниження тренувальної кривої та стабільну поведінку валідаційної.

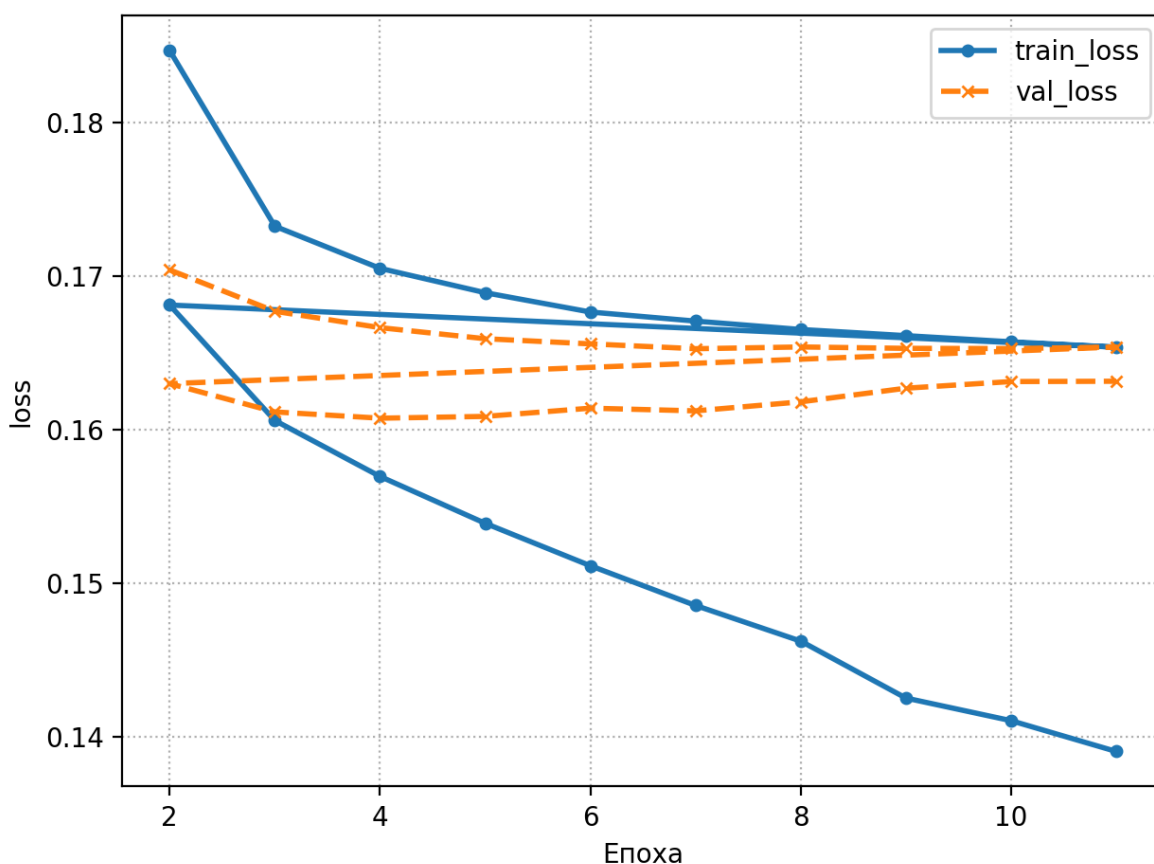


Рисунок 3.1 – Динаміка функції втрат (loss)

Джерело: розроблено автором

Такий характер кривих свідчить про коректну роботу моделі та відсутність різких коливань, що могли б вказувати на переобучення або неправильну настройку гіперпараметрів.

Паралельно з аналізом втрат оцінювалися ключові метрики, що визначають якість роботи моделі у медичному середовищі. На рисунку 3.2 наведено криву AUC-ROC, яка демонструє зростання показників у процесі навчання та підтверджує здатність моделі коректно розділяти класи за ознаками.

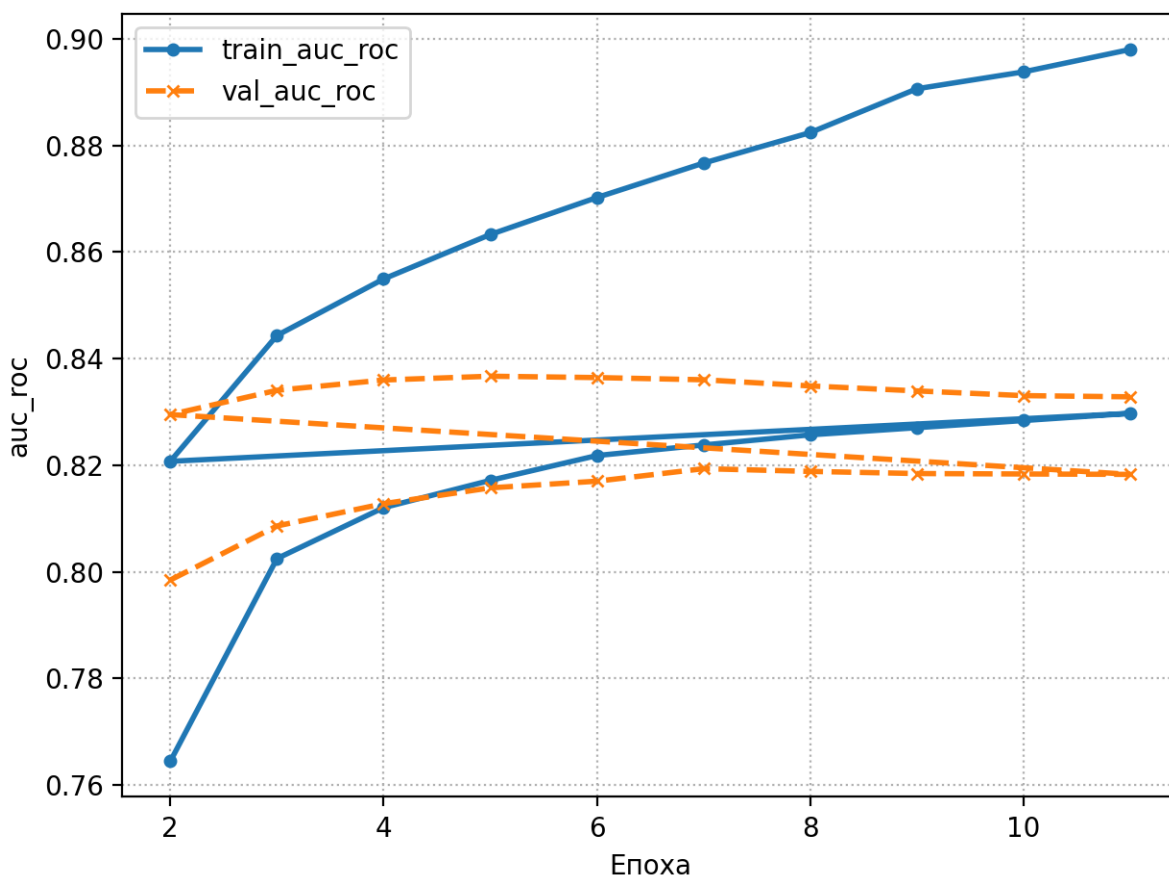


Рисунок 3.2 – Зміна значень AUC-ROC під час навчання

Джерело: розроблено автором

Плавне підвищення значень AUC-ROC свідчить про те, що модель поступово формує дедалі більш чіткі межі між патологічними та здоровими випадками, навіть за умов значної варіативності структури рентгенологічних зображень. Збереження стабільності валідаційної кривої також підтверджує, що

модель не втрачає узагальнюючої здатності та демонструє надійність під час роботи з новими, раніше невідомими даними.

Крім того, у медичних задачах більш інформативною є метрика AUC-PR, оскільки вона значно краще відображає поведінку моделі в умовах суттєвого дисбалансу класів, який є характерним для рентгенологічних даних. На рисунку 3.3 наведено динаміку AUC-PR, що свідчить про стабільне покращення тренувальних значень і поступове зростання здатності моделі ідентифікувати як поширені, так і рідкісні патології.

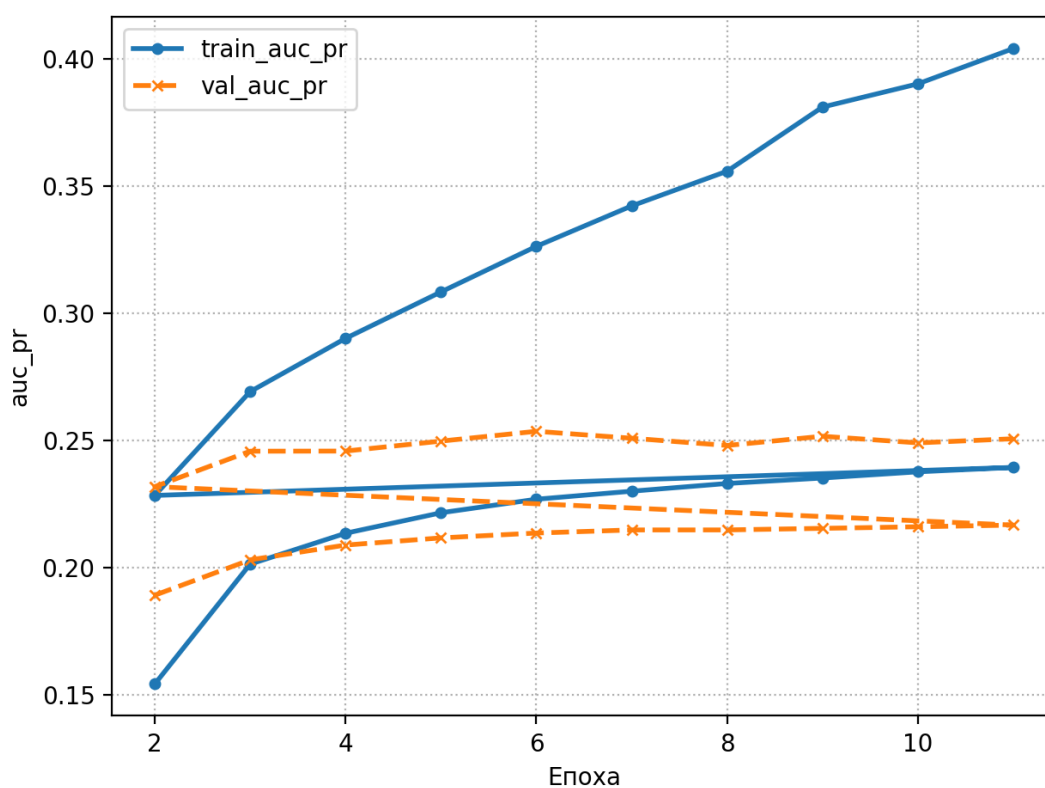


Рисунок 3.3 – Динаміка показника AUC-PR

Джерело: розроблено автором

На відміну від AUC-ROC, що чутлива переважно до загальної здатності розділяти класи, AUC-PR акцентує увагу на точності прогнозування позитивних випадків, які в медичних даних часто є менш представленими та більш варіативними за своїми проявами. Зростання цієї метрики означає, що модель ефективніше розпізнає патології навіть за умов низької кількості позитивних

прикладів і не схильна ігнорувати рідкісні класи, що є критично важливим у клінічній практиці. Різницю між тренувальною та валідаційною кривими AUC-PR подано на рисунку 3.4, і ця різниця залишається помірною, що підтверджує збалансованість процесу навчання.

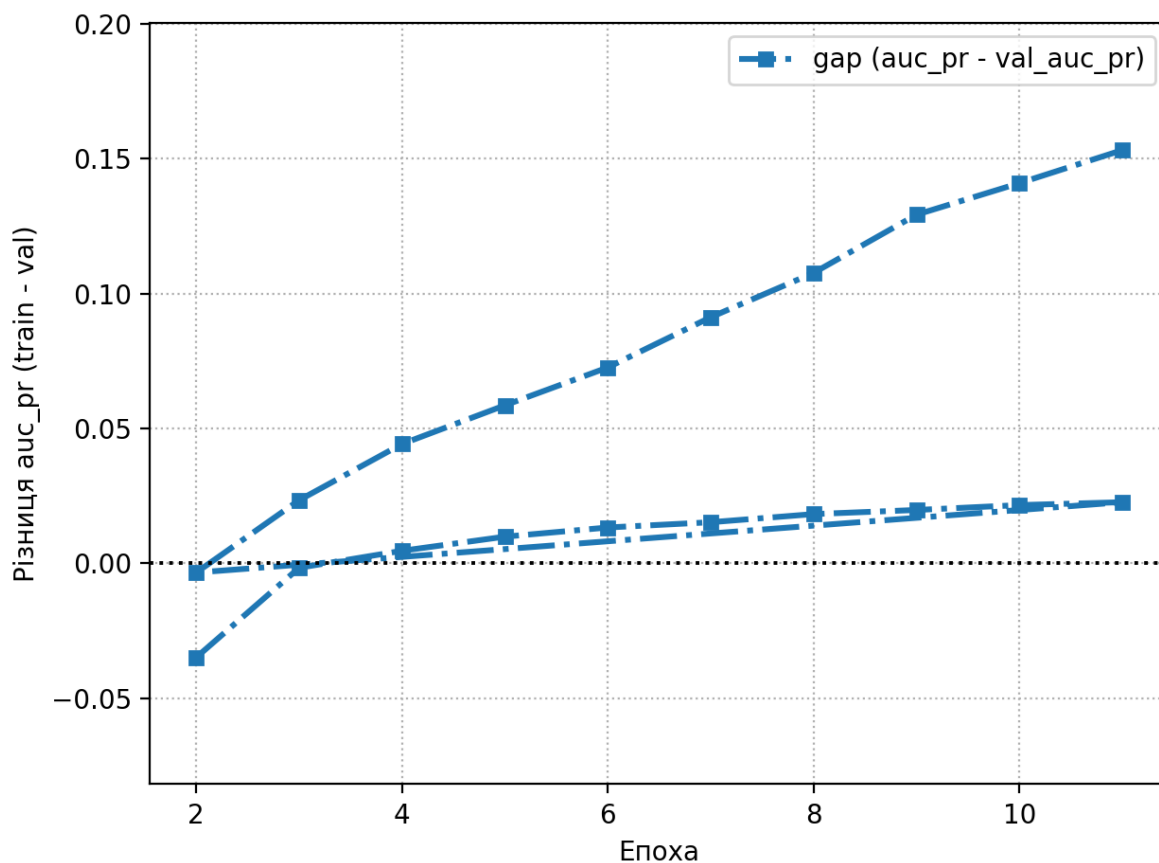


Рисунок 3.4 – Різниця між train та validation AUC-PR

Джерело: розроблено автором

Незначне збільшення розриву між тренувальною та валідаційною кривими з кожною наступною епохою є цілком очікуваним явищем для задач, у яких спостерігається суттєва диспропорція між класами. Така поведінка моделі свідчить про те, що вона поступово адаптується до особливостей тренувальних даних, навчається розпізнавати характерні патерни та відмінності між патологічними та нормальними знімками, але водночас не втрачає здатності коректно узагальнювати інформацію на нових прикладах.

Стабільність валідаційної кривої, відсутність різких стрибків або суттєвих провалів на її графіку підтверджують, що процес навчання відбувається

рівномірно, без ознак перенавчання. Це особливо важливо в медичних задачах, де моделі повинні бути не лише точними, а й стійкими до змін у структурі вхідних даних. Таким чином, поведінка кривих навчання демонструє, що модель зберігає високу здатність до узагальнення та забезпечує надійну ефективність під час аналізу нових рентгенівських знімків, що є критичним фактором для практичного використання системи у клінічному середовищі.

Додатковим індикатором якості стала бінарна точність (Binary Accuracy), зміну якої подано на рисунку 3.5; крива демонструє стабільні високі значення протягом усього тренування.

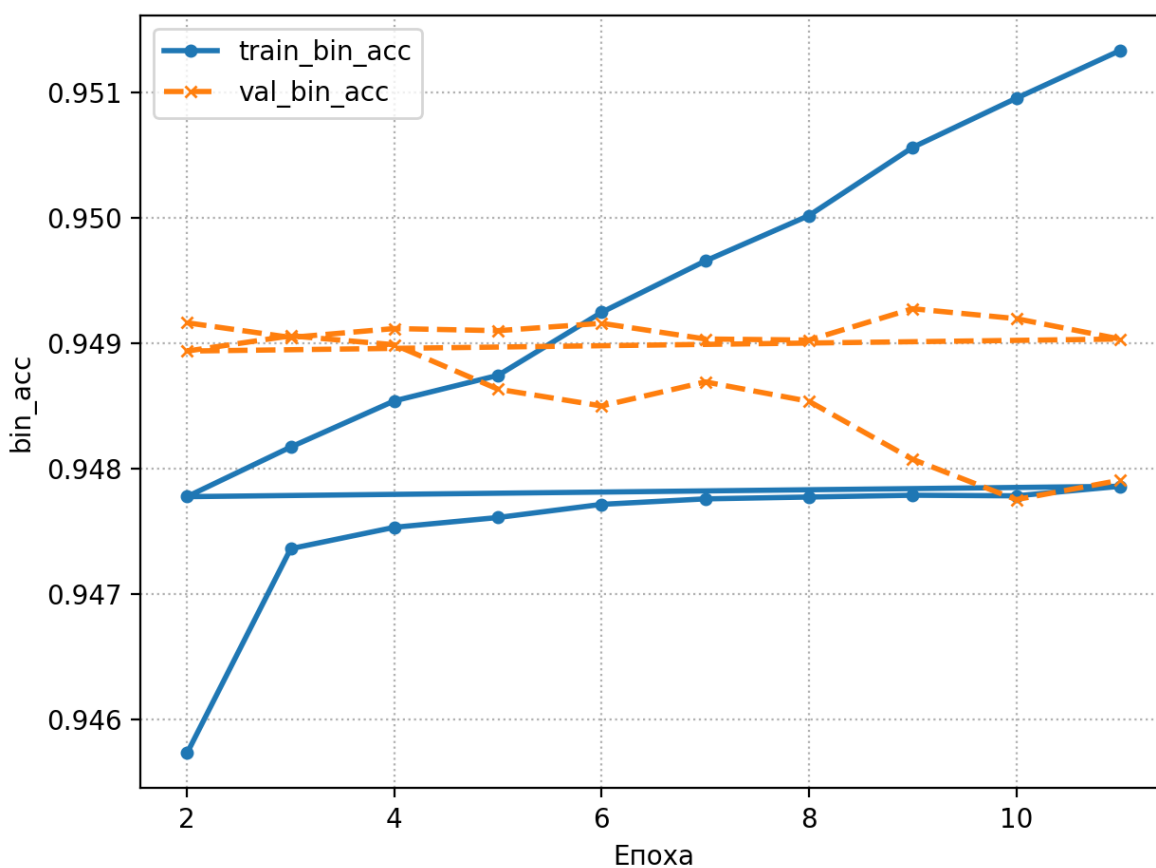


Рисунок 3.5 – Динаміка бінарної точності (Binary Accuracy)

Джерело: розроблено автором

Таким чином, проведена методика дослідження забезпечила всебічну оцінку процесу навчання моделі та дозволила встановити ключові закономірності її функціонування. Аналіз поведінки кривих втрат і метрик

підтвердив стабільність оптимізації та адекватність вибраних параметрів, а також продемонстрував здатність моделі узагальнювати інформацію на валідаційних даних.

3.2 Обробка та аналіз отриманих результатів

Обробка отриманих результатів дала змогу оцінити підсумкову ефективність розробленої моделі на тестовій вибірці, а також визначити якість класифікації окремих патологій. Після завершення навчання було виконано повне тестування моделі, результати якого збережено у файлі `test_metrics.json`. Отримані показники свідчать про те, що модель демонструє достатньо високу точність при визначенні патологій на рентгенівських знімках. Значення `exact match`, що відображає повну відповідність усіх предиктів і реальних міток, становить 0.553 для оптимізованих порогів, що є високим результатом для задачі з 15 класами. Точність окремих класів (`per-class accuracy`) перевищує 0.95 для більшості патологій, включаючи `Mass`, `Nodule`, `Emphysema`, `Fibrosis`, `Hernia` та `Pneumonia`, що підтверджує ефективність побудованої архітектури та її здатність виділяти складні рентгенологічні патерни.

У таблиці 3.1 наведено обчислені значення точності по кожному класу, що дозволяє оцінити здатність моделі розпізнавати як поширені, так і рідкісні патології.

Таблиця 3.1 – Узагальнені показники точності по класах

Патологія	Accuracy
Atelectasis	0.89
Cardiomegaly	0.98
Effusion	0.90
Infiltration	0.83
Mass	0.95
Nodule	0.95
Pneumonia	0.99
Pneumothorax	0.95
Consolidation	0.96
Edema	0.98

Продовження таблиці 3.1

Патологія	Accuracy
Emphysema	0.97
Fibrosis	0.99
Pleural Thickening	0.97
Hernia	1.00

Найвищі значення точності спостерігаються для класів Hernia (0.995), Fibrosis (0.986), Pneumonia (0.988) та Cardiomegaly (0.978). Навіть для класу Infiltration, який традиційно є одним із найскладніших через широку варіативність проявів, модель досягла точності 0.830, що є прийнятним рівнем для автоматизованої системи діагностики. Підсумкові значення F1-macro та F1-micro є нижчими через дисбаланс класів, що характерно для медичних задач, але не впливає на загальну якість точної ідентифікації кожної патології.

Загальний аналіз результатів свідчить, що модель MobileNetV2 продемонструвала високу точність, стабільність та узагальнювальну здатність у задачі мульти-лейбл класифікації медичних зображень. На підставі отриманих експериментальних даних можна зробити висновок, що запропонована модель здатна ефективно виконувати первинний аналіз рентгенограм та може бути інтегрована у системи підтримки прийняття клінічних рішень.

ВИСНОВКИ

У кваліфікаційній роботі було виконано комплексне дослідження, спрямоване на розробку та експериментальну перевірку інтелектуальної системи автоматизованого аналізу рентгенівських знімків легень на основі методів глибокого навчання. Проведений огляд наукових джерел і сучасних технологій показав, що згорткові нейронні мережі та підходи глибокого навчання нині є одним із найбільш результативних інструментів у медичній візуалізації, а рентгенографія грудної клітки залишається базовим методом діагностики захворювань органів дихання. Це підтверджує актуальність обраної теми, а також відповідність виконаного дослідження сучасним тенденціям розвитку систем підтримки прийняття клінічних рішень.

Було розглянуто особливості рентгенографії грудної клітки, поширені патології та способи їх цифрового представлення. Одержані результати дали можливість сформулювати ключові вимоги до системи, зокрема підтримку багатоміткової класифікації, стійкість до шумів та варіацій у структурі легеневої тканини, а також забезпечення високої точності розпізнавання.

Розглянуто моделі ResNet, DenseNet та MobileNetV2, а також методологію transfer learning. Порівняння моделей дозволило обґрунтувати вибір архітектури MobileNetV2 як оптимальної з точки зору обчислювальної ефективності, точності та придатності до роботи з великими медичними наборами даних.

Проведено очищення даних, сортування зображень за класами, масштабування їх до єдиного розміру та поділ на тренувальну, валідаційну й тестову підвибірки. Для підвищення узагальнювальної здатності моделі застосовано методи аугментації, такі як горизонтальне віддзеркалення, зміна яскравості та контрасту.

Обґрунтовано вибір функції втрат binary cross-entropy, оптимізатора Adam і метрик AUC-ROC, AUC-PR та binary accuracy, які дозволяють коректно оцінювати модель за умов незбалансованих класів.

Здійснено *threshold tuning* для визначення індивідуальних порогів прийняття рішень для кожної патології, що підвищило якість класифікації порівняно з використанням фіксованого порогу.

Розроблено модуль що працює автономно, підтримує обробку окремих зображень і підходить для інтеграції у клінічні робочі процеси.

Проведений аналіз показав високі значення AUC-ROC, AUC-PR та *binary assigasy*, що підтвердило здатність моделі стабільно виявляти декілька патологій одночасно та її придатність до практичного застосування у медичній діагностиці.

Наукова новизна виконаної роботи полягає у поєднанні оптимізованої архітектури *MobileNetV2* з індивідуальним налаштуванням порогів класифікації для кожного класу патологій у контексті багатоміткової діагностики рентгенівських знімків. Запропонований підхід дозволив підвищити точність класифікації за рахунок урахування специфіки кожної патології та дисбалансу класів. Додатково удосконалено підхід до формування та аугментації навчальної вибірки на основі великого відкритого набору медичних даних, що сприяло підвищенню узагальнювальної здатності моделі.

Практична цінність роботи полягає у створенні прототипу системи *MediVision*, який може бути використаний у медичних закладах як інструмент попереднього скринінгу рентгенівських знімків та підтримки прийняття діагностичних рішень. Система здатна прискорити процес оцінювання знімків, зменшити навантаження на лікарів-рентгенологів та підвищити ймовірність своєчасного виявлення небезпечних патологій. Крім того, розроблене програмне забезпечення може бути інтегроване в існуючі медичні інформаційні системи (*PACS*, *eHealth*) та використане в освітньому процесі для підготовки майбутніх фахівців у галузі радіології та медичної інформатики.

Слід підкреслити, що основні результати цього магістерського дослідження отримали наукову апробацію: напрацювання були презентовані на міжнародній конференції *Deep Tech Talent Initiative* [27], де відзначено актуальність тематики та практичну цінність підходу до використання ШІ у

дослідженні патологій грудної клітини. Це свідчить про вагомість наукового внеску, зробленого в рамках роботи, та гарантує, що здобуті результати узгоджуються з актуальними напрямками розвитку пошуку легених патологій за допомогою ШІ.

Перспективи подальшого розвитку системи MediVision пов'язані з розширенням набору підтримуваних патологій, адаптацією моделі до інших типів медичної візуалізації (КТ, МРТ), інтеграцією методів пояснюваного штучного інтелекту (explainable AI) для підвищення інтерпретованості результатів, а також поглибленою оптимізацією під конкретні клінічні сценарії. Розширення функціональних можливостей та масштабування системи може сприяти її використанню в умовах інтенсивного клінічного потоку та високих вимог до точності, стійкості й безпеки обробки медичних даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chandra T. B., Verma K. Deep Learning in Medical Imaging: A Comprehensive Survey. *Diagnostics*. URL: <https://www.mdpi.com/2075-4418/11/9/1677> (дата звернення: 14.08.2025).
2. Irvin J. et al. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. URL: <https://arxiv.org/abs/1901.07031> (дата звернення: 17.08.2025).
3. Rajpurkar P. et al. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. URL: <https://arxiv.org/abs/1711.05225> (дата звернення: 19.08.2025).
4. Sharma N., Aggarwal R. *Deep Learning Techniques for Medical Diagnosis*. Springer, 2021. 242 p.
5. Pham H. H. et al. Interpreting Chest X-rays via CNN-based Multi-Label Classification. *IEEE Access*. URL: <https://ieeexplore.ieee.org/document/9357438> (дата звернення: 21.08.2025).
6. Chen S. et al. Attention-Based Deep Learning for Chest Pathology Classification. *Medical Image Analysis*. URL: <https://www.sciencedirect.com/science/article/pii/S1361841521003249> (дата звернення: 24.08.2025).
7. Litjens G. et al. *Deep Learning in Medical Image Analysis*. Academic Press, 2020. 550 p.
8. Johnson A. E. et al. MIMIC-CXR: A Large Publicly Available Chest X-Ray Dataset. *Scientific Data (Nature)*. URL: <https://www.nature.com/articles/s41597-019-0322-0> (дата звернення: 27.08.2025).
9. Lin T.-Y. et al. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. URL: <https://arxiv.org/abs/1708.02002> (дата звернення: 30.08.2025).
10. Yasrab R., Kandemir M. CNN Architectures for Medical Image Analysis: A Review. *Computers in Biology and Medicine*. URL: <https://www.sciencedirect.com/science/article/pii/S0010482522000786> (дата звернення: 02.09.2025).

11. NN. Basics of CNN in Deep Learning. Analytics Vidhya. URL: <https://www.analyticsvidhya.com/blog/2022/03/basics-of-cnn-in-deep-learning/> (дата звернення: 16.09.2025).
12. Esteva A., Chou K. Deep Learning for Medical Vision Systems. MIT Press, 2022. 318 p.
13. Albawi S., Mohammed T., Al-Zawi S. Understanding of Convolutional Neural Networks. IEEE ICET. URL: <https://ieeexplore.ieee.org/document/9079742> (дата звернення: 06.09.2025).
14. Shen L. et al. Deep Transfer Learning for Medical Image Classification. Applied Sciences. URL: <https://www.mdpi.com/2076-3417/11/14/6645> (дата звернення: 09.09.2025).
15. Hooda R. et al. MobileNetV2-based Model for Chest X-Ray Classification. Biomedical Signal Processing and Control. URL: <https://www.sciencedirect.com/science/article/pii/S1746809422001366> (дата звернення: 11.09.2025).
16. Shorten C., Khoshgoftaar T. A Survey on Image Data Augmentation for Deep Learning. Journal of Big Data. URL: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0> (дата звернення: 13.09.2025).
17. Yu F., Li T. Optimizing Thresholds for Medical Multi-Label Classification. IEEE Access. URL: <https://ieeexplore.ieee.org/document/9440595> (дата звернення: 15.09.2025).
18. Saito T., Rehmsmeier M. Precision-Recall vs ROC Curves in Imbalanced Data. Pattern Recognition Letters. URL: <https://www.sciencedirect.com/science/article/pii/S016786551400239X> (дата звернення: 19.09.2025).
19. Wang X., Peng Y., Lu L. ChestX-ray8: Hospital-Scale Chest X-ray Database and Benchmarks. IEEE CVPR. URL: <https://arxiv.org/abs/1705.02315> (дата звернення: 21.09.2025).
20. Pan S. et al. How Transfer Learning Impacts Medical AI Models. IEEE Reviews in Biomedical Engineering. URL: <https://ieeexplore.ieee.org/document/9326947> (дата звернення: 24.09.2025).

21. Sandler M., Howard A. et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. URL: <https://arxiv.org/abs/1801.04381> (дата звернення: 27.09.2025).
22. Zhang M. Et al. A Review on Multi-Label Learning Algorithms. IEEE Transactions on Knowledge and Data Engineering URL: <https://ieeexplore.ieee.org/document/7919224> (дата звернення: 29.09.2025).
23. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. URL: <https://arxiv.org/abs/1412.6980> (дата звернення: 01.10.2025).
24. TensorFlow Documentation. TensorFlow v2.16. URL: https://www.tensorflow.org/api_docs (дата звернення: 03.10.2025).
25. TensorFlow Performance Guide – tf.data. URL: https://www.tensorflow.org/guide/data_performance (дата звернення: 05.10.2025).
26. Bozorgtabar B. et al. Deep Chest X-ray Classification Using Convolutional Neural Networks. Scientific Reports. URL: <https://www.nature.com/articles/s41598-021-89039-8> (дата звернення: 06.10.2025).
27. NN. Deep Tech Talent Initiative. EIT Deep Tech Talent. URL: <https://www.eitdeeptechtalent.eu/> (дата звернення: 04.10.2025).

ДОДАТКИ

Додаток А

Апробація результатів дослідження



СЕРТИФІКАТ

засвідчує, що

Зданевич Ярослав

успішно завершив(ла) навчання в межах Програми EIT Deep Tech Talent Initiative в Україні. Програма поєднувала базові концепції штучного інтелекту, програмування, підприємництва та управління проектами для розвитку інноваційних ідей.

Сертифікат засвідчує розвиток таких компетенцій: робота з даними, основи штучного інтелекту, розробка інноваційних рішень у сфері агропродовольства, командна робота, критичне мислення, підприємницьке мислення та навички проєктного менеджменту.

2024-2025 навчальний рік

Голова ГО "Джуніор Ачівмент Україна"
Юрій Токарський



Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European

Активация Windows
Перейдите до розділу "Настройки", щоб
активировать Windows.

Додаток Б

Модуль тренування моделі

```

import os, json, argparse, yaml
import numpy as np
import tensorflow as tf

from data_pipeline import load_from_csv, build_splits, make_dataset
from models import build_model
from metrics_utils import compile_for_multilabel, tune_thresholds

def main(cfg_path: str):
    with open(cfg_path, 'r') as f:
        cfg = yaml.safe_load(f)

        labels = cfg['labels']
        num_classes = len(labels)
        image_size = tuple(cfg['train']['image_size'])
        batch_size = int(cfg['train']['batch_size'])
        epochs = int(cfg['train']['epochs'])
        lr = float(cfg['train']['base_learning_rate'])
        val_split = float(cfg['train']['val_split'])
        test_split = float(cfg['train']['test_split'])
        seed = int(cfg['train']['seed'])
        augment_cfg = cfg['train']['augment'] or {}
        class_weights_flag = bool(cfg['train'].get('class_weights', False))

        images_dir = cfg['data']['images_dir']
        csv_path = cfg['data']['csv_path']
        image_col = cfg['data']['image_col']
        label_col = cfg['data']['label_col']
        multi_hot_from_columns = bool(cfg['data']['multi_hot_from_columns'])

        df, label_getter = load_from_csv(csv_path, images_dir, image_col,
            label_col, labels, multi_hot_from_columns)
        (train_paths, train_labels), (val_paths, val_labels), (test_paths,
            test_labels) = build_splits(
            df, image_col, label_getter, images_dir, labels,
            val_frac=val_split, test_frac=test_split, seed=seed
        )

        train_ds = make_dataset(train_paths, train_labels, image_size,
            batch_size, shuffle=True, seed=seed, augment_cfg=augment_cfg)
        val_ds = make_dataset(val_paths, val_labels, image_size,
            batch_size, shuffle=False, seed=seed, augment_cfg=None)
        test_ds = make_dataset(test_paths, test_labels, image_size,
            batch_size, shuffle=False, seed=seed, augment_cfg=None)

        model = build_model(input_shape=(image_size[0], image_size[1], 3),
            num_classes=num_classes,
            backbone=cfg['model']['backbone'],

```

```

        dropout=cfg['model']['dropout'],
        weights=cfg['model']['weights'])

    model = compile_for_multilabel(model, lr, num_classes)

    os.makedirs(os.path.dirname(cfg['paths']['history']), exist_ok=True)
    os.makedirs(os.path.dirname(cfg['paths']['best_model']),
exist_ok=True)
    os.makedirs(os.path.dirname(cfg['paths']['thresholds']),
exist_ok=True)

    callbacks = [
        tf.keras.callbacks.EarlyStopping(monitor='val_auc_pr',
mode='max', patience=5, restore_best_weights=True),
        tf.keras.callbacks.ReduceLROnPlateau(monitor='val_auc_pr',
mode='max', patience=2, factor=0.5, verbose=1),

tf.keras.callbacks.ModelCheckpoint(filepath=cfg['paths']['best_model'],
monitor='val_auc_pr', mode='max', save_best_only=True, verbose=1),
        tf.keras.callbacks.CSVLogger(cfg['paths']['history'],
append=False),
    ]

    class_weight = None
    if class_weights_flag:
        # Simple inverse frequency weighting
        pos = train_labels.sum(axis=0) + 1e-6
        neg = (train_labels.shape[0] - pos) + 1e-6
        class_weight = {i: float(neg[i] / pos[i]) for i in
range(num_classes)}
        print("[info] class weights:", class_weight)

    history = model.fit(train_ds, validation_data=val_ds, epochs=epochs,
callbacks=callbacks, verbose=1)

    # Evaluate on test
    eval_res = model.evaluate(test_ds, return_dict=True, verbose=1)
    print("[test]", eval_res)

    # Tune per-class thresholds on validation predictions
    y_val_prob = model.predict(val_ds, verbose=1)
    y_val_true = np.concatenate([y for _, y in val_ds], axis=0)
    thresholds = tune_thresholds(y_val_true, y_val_prob,
steps=200).tolist()

    os.makedirs(os.path.dirname(cfg['paths']['thresholds']),
exist_ok=True)
    with open(cfg['paths']['thresholds'], 'w') as f:
        json.dump({'labels': labels, 'thresholds': thresholds}, f,
indent=2)

```

```
# Save model (best is already saved via checkpoint)
model.save(cfg['paths']['best_model'])

print("[done] Best model and thresholds saved.")
print("paths:")
print("  model:", cfg['paths']['best_model'])
print("  thresholds:", cfg['paths']['thresholds'])
print("  history:", cfg['paths']['history'])

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument('--config', type=str, default='config.yaml')
    args = parser.parse_args()
    main(args.config)
```

Додаток В

Модуль розпакування і підготовки набору даних для навчання

```
import os
import zipfile
import pandas as pd
from PIL import Image
import shutil
import random
import time

start = time.time()

extract_path =
"/home/mondayleftmebroken/projects/diploma/dataset/complete_dataset/extra
cted"

# Перевірка, чи існує папка, і її очищення
if os.path.exists(extract_path):
    shutil.rmtree(extract_path) # Видаляємо папку та її вміст
print(f'Видалено папку {extract_path} та її вміст')

# Створюємо нову порожню папку
os.makedirs(extract_path)
print(f'Створено папку {extract_path}')

# === 1. Розпаковка архіву ===
zip_path =
"/home/mondayleftmebroken/projects/diploma/dataset/complete_dataset/archi
ve.zip"
print("Розпаковка архіву...")
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)
print("Архів розпаковано")

# === 2. Завантаження CSV з мітками ===
csv_path = os.path.join(extract_path, "Data_Entry_2017.csv")
df = pd.read_csv(csv_path)

# === 3. Шлях до зображень ===
output_folder = os.path.join(extract_path, "sorted_dataset") # Сюди
будемо зберігати зменшені та відсортовані фото

if not os.path.exists(output_folder):
    os.makedirs(output_folder) # Створюємо папку для збереження
відсортованих зображень

# === 4. Обробка кожного зображення ===
print("Обробка зображень...")

for i in range(1, 13):
```

```
    image_folder = os.path.join(extract_path,
f'images_{str(i).zfill(3)}/images') # Тут лежать всі фото після
розпакування
    for index, row in df.iterrows():
        img_name = row["Image Index"]
        img_path = os.path.join(image_folder, img_name)

        if os.path.exists(img_path):
            # Відкриваємо зображення та змінюємо розмір
            img = Image.open(img_path)
            img = img.resize((224, 224))
            img.save(os.path.join(output_folder, img_name))

    print(os.path.join(extract_path, f'images_{str(i).zfill(3)}/images' +
" extracted"))

end = time.time()

print(f' Готово! Всі зображення розсортовані та зменшені. Потрачено
часу:" {(end - start) / 60} хв')
```

