Міністерство освіти і науки України
Луцький національний технічний університет
Факультет робототехніки та штучного інтелекту
Кафедра штучного інтелекту та математичного моделювання

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ
«БАКАЛАВР»

ДОСЛІДЖЕННЯ ТА РОЗРОБКА МЕТОДІВ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ РЕНТГЕНІВСЬКИХ ЗОБРАЖЕНЬ ДЛЯ ВИЯВЛЕННЯ ПНЕВМОНІЇ

RESEARCH AND DEVELOPMENT OF AUTOMATIC CLASSIFICATION METHODS FOR X-RAY IMAGES FOR PNEUMONIA DETECTION

Спеціальність 113 Прикладна математика
(шифр і назва спеціальності)

освітня програма «Штучний інтелект та аналіз масивів даних»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ПРМ-41
Перепич Анна Олексіївна
_____
(підпис)

Керівник:
к.т.н., доцент
Приходько Олексій Сергійович
_____
(підпис)

Кваліфікаційну роботу
допущено до захисту
«___» _____ 20__ р.
к.т.н., доцент
Гарант освітньої програми:
Приходько Олексій Сергійович
_____
(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *архітектури, будівництва та дизайну*
Кафедра *прикладної математики та механіки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *11 Математика і статистика*
Спеціальність *113 Прикладна математика*
Освітня програма *Штучний інтелект та аналіз масивів даних*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____Мікуліч О.А.
«___» _____2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Перепич Анна Олексіївна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Дослідження та розробка методів автоматичної класифікації рентгенівських зображень для виявлення пневмонії | Research and development of automatic classification methods for X-ray images for pneumonia detection
Керівник роботи: *Приходько Олексій Сергійович*
затверджені наказом закладу вищої освіти від «31» грудня 2025 р. № 557/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 04.06.2026 р.

3. Вихідні дані до роботи *профільні публікації з штучного інтелекту та математичного моделювання в межах досліджуваної проблематики; релевантні набори даних; математичні моделі цільових процесів; технічна документація Python-бібліотек та методичні вказівки до виконання кваліфікаційної роботи бакалавра*

4. Зміст пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз предметної області
Постановка задачі та вибір методів
Практична реалізація
Отримання та аналіз результатів

5. Перелік графічного (ілюстративного) матеріалу:
Презентація роботи (слайди): об'єкт, предмет, мета та завдання дослідження; аналіз предметної області; математичні та алгоритмічні моделі, результати експериментальних досліджень (метрики та візуалізація роботи алгоритму); загальні висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
1 розділ	*Приходько О.С., доцент кафедри*		
2 розділ	*Приходько О.С., доцент кафедри*		
3 розділ	*Приходько О.С., доцент кафедри*		
4 розділ	*Приходько О.С., доцент кафедри*		
Висновки	*Приходько О.С., доцент кафедри*		

7. Дата видачі завдання «___» _____ 202__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1.	*Огляд літератури із досліджуваної проблеми*	*до 01.03.2026*	
2.	*Перший розділ*	*до 5.03.2026*	
3.	*Другий розділ*	*до 01.04.2026*	
4.	*Третій розділ*	*до 10.04.2026*	
5.	*Четвертий розділ*	*до 20.04.2026*	
6.	*Висновки*	*до 28.04.2026*	
7.	*Формування списку використаних джерел*	*до 05.05.2026*	
8.	*Оформлення ілюстративного матеріалу*	*до 09.05.2026*	
9.	*Нормоконтроль*	*до 20.05.2026*	
10.	*Інструментальна перевірка на академічний плагіат*	*до 02.062026*	*Показник запозичень тексту ____ %*
11.	*Представлення кваліфікаційної роботи бакалавра до захисту*	*до 04.06.2026*	

Здобувач вищої освіти _____ (Перепич А. О.)
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ (Приходько. О. С.)
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Перепич А. О. Дослідження та розробка методів автоматичної класифікації рентгенівських зображень для виявлення пневмонії. Рукопис.

Кваліфікаційна робота бакалавра ОП «Штучний інтелект та аналіз масивів даних» спеціальності 113 Прикладна математика. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота присвячена розробці та дослідженню методів комп'ютерного зору для автоматизованої діагностики захворювань легень за двовимірними рентгенівськими знімками. В процесі роботи проаналізовано масив даних COVID-19 Radiography Database (понад 21 тис. зображень). Виявлено проблему структурного міжкласового дисбалансу, для вирішення якої математично обґрунтовано використання макро-усередненої метрики повноти (Macro-Recall) та зваженої функції втрат багатокласової перехресної ентропії.

Проведено порівняльний експериментальний аналіз базової згорткової нейронної мережі (CNN) та моделі на основі глибокої архітектури ResNet50 із застосуванням парадигми перенесення навчання.

Для подолання проблеми «чорного ящика» успішно імплементовано алгоритм просторової локалізації Grad-CAM. Аналіз отриманих теплових карт підтвердив, що розроблена нейромережева система приймає рішення на основі релевантних просторових клінічних маркерів (вогнищ консолідації та зон «матового скла»). Одержані результати доводять високу предиктивну здатність розробленої системи та її придатність для використання у якості асистента підтримки прийняття медичних рішень.

Ключові слова: комп'ютерний зір, медична візуалізація, згорткові нейронні мережі, перенесення навчання, пневмонія, COVID-19, дисбаланс класів, Grad-CAM.

ABSTRACT

Perepych A. O. Research and development of automatic classification methods for X-ray images for pneumonia detection. Manuscript.

Bachelor's Thesis in the field of "Artificial Intelligence and BigData Analysis" in specialty 113 Applied Mathematics. Lutsk National Technical University. Lutsk, 2026.

The bachelor's thesis is devoted to the development and research of computer vision methods for the automated diagnosis of lung diseases using two-dimensional chest X-ray images. During the research, the COVID-19 Radiography Database (over 21,000 images) was analyzed. The problem of structural interclass imbalance was identified, for the solution of which the use of the macro-averaged recall metric (Macro-Recall) and the weighted categorical cross-entropy loss function was mathematically justified.

A comparative experimental analysis of a baseline convolutional neural network (CNN) and a model based on the deep ResNet50 architecture using the Transfer Learning paradigm was conducted. The empirical superiority of the pre-trained model was proven, which allowed solving the problem of high algorithmic variance, avoiding overfitting, and achieving an overall classification accuracy of 91.52% with a pathology detection macro-recall of 91.21%.

To overcome the "black box" problem, the Grad-CAM spatial localization algorithm was successfully implemented. The analysis of the obtained heatmaps confirmed that the developed neural network system makes decisions based on relevant spatial clinical markers (consolidation foci and "ground-glass" opacities). The obtained results prove the high predictive ability of the developed system and its suitability for use as a medical clinical decision support assistant.

Keywords: computer vision, medical imaging, convolutional neural networks, transfer learning, pneumonia, COVID-19, class imbalance, Grad-CAM.

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1 10
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ДАНИХ 10
1.1. Проблематика діагностики захворювань легень за допомогою рентгенографії 10
1.2. Огляд існуючих методів комп'ютерного зору в медичній візуалізації . 11
1.3. Аналіз доступних масивів рентгенівських зображень 12
Висновки до розділу 1 14
РОЗДІЛ 2 16
ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР МЕТОДІВ РОЗВ'ЯЗАННЯ 16
2.1. Математична постановка задачі класифікації медичних зображень 16
2.2. Вибір метрик для оцінки ефективності моделей в умовах дисбалансу класів 17
2.3. Обґрунтування вибору архітектур згорткових нейронних мереж..... 19
2.4. Методи підвищення інтерпретованості рішень штучного інтелекту в медицині..... 20
Висновки до розділу 2 21
РОЗДІЛ 3 22
РОЗРОБКА ТА ІМПЛЕМЕНТАЦІЯ РІШЕННЯ..... 22
3.1. Загальна архітектура системи автоматичної класифікації зображень .. 22
3.2. Попередня обробка та аугментація масивів візуальних даних 23
3.3. Розробка базової моделі згорткової нейронної мережі 25
3.4. Програмна реалізація методів перенесення навчання 27
Висновки до розділу 3 29
РОЗДІЛ 4 31
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ..... 31
4.1. Методика проведення експериментів та налаштування гіперпараметрів 31
4.2. Аналіз результатів роботи базової моделі класифікації 32
4.3. Аналіз результатів застосування попередньо навчених моделей..... 33
4.4. Візуалізація та інтерпретація ознак за допомогою методу Grad-CAM. 35
Висновки до розділу 4 36
ВИСНОВКИ..... 38

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 40
ДОДАТКИ 41
Додаток А Лістинг програмного модуля попередньої обробки та розвідувального аналізу даних 41
Додаток Б Лістинг програмного модуля оптимізації базової згорткової нейронної мережі 42
Додаток В Лістинг програмного модуля класифікації на основі Transfer Learning (ResNet50) 44
Додаток Г Лістинг програмного модуля інтерпретації результатів (Grad-CAM) 47

ВСТУП

Актуальність теми. Своєчасна та точна діагностика респіраторних захворювань, зокрема вірусних пневмоній та COVID-19, є однією з пріоритетних задач сучасної медицини. Рентгенографія органів грудної клітки залишається найбільш доступним інструментом первинного скринінгу. Однак аналіз таких медичних зображень ускладнюється інтенсивним структурним шумом (накладанням тривимірних анатомічних утворень на двовимірну площину), високим ступенем візуальної подібності різних типів легеневих патологій та суб'єктивним людським фактором, що виникає через когнітивну втому лікарів. Впровадження методів штучного інтелекту, зокрема глибоких згорткових нейронних мереж (CNN), здатне суттєво підвищити швидкість та об'єктивність діагностики. Разом з тим, застосування базових алгоритмів машинного навчання часто стикається з проблемами структурного дисбалансу медичних масивів даних, ефектом перенавчання та феноменом «чорного ящика». Це зумовлює об'єктивну необхідність дослідження та розробки адаптованих нейромережевих архітектур, заснованих на парадигмі перенесення навчання з механізмами просторової інтерпретації результатів. Усе вищезазначене підтверджує високу актуальність обраної теми кваліфікаційної роботи.

Мета і завдання дослідження. Метою роботи є підвищення точності та надійності автоматизованої діагностики патологій легень шляхом розробки, математичного обґрунтування та програмної імплементації методів комп'ютерного зору для класифікації рентгенівських зображень. Для досягнення поставленої мети було сформульовано наступні завдання:

1. Провести розвідувальний аналіз відкритого масиву медичних даних та визначити його статистичні характеристики.
2. Здійснити математичну постановку задачі оптимізації та обґрунтувати вибір метрик оцінювання ефективності в умовах дисбалансу класів.

3. Розробити конвеєр попередньої обробки просторових візуальних даних із застосуванням методів аугментації та зважування класів.

4. Спроектувати та дослідити базову архітектуру згорткової нейронної мережі для встановлення емпіричного рівня продуктивності.

5. Розробити модель на базі архітектури ResNet50 із застосуванням парадигми Transfer Learning.

6. Провести порівняльний експериментальний аналіз моделей та забезпечити клінічну інтерпретованість результатів за допомогою алгоритму Grad-CAM.

Об’єкт дослідження – процес автоматичної класифікації медичних візуальних даних.

Предмет дослідження – математичні моделі, методи глибокого навчання та алгоритми виділення візуальних ознак на рентгенівських зображеннях органів грудної клітки.

Методи дослідження. У роботі використано методи теорії ймовірностей та математичної статистики (для аналізу датасету та вибору метрик), методи нелінійної оптимізації (для мінімізації функції втрат градієнтним спуском), методи глибокого машинного навчання (згорткові нейронні мережі) та методи Explainable AI (просторова візуалізація градієнтів Grad-CAM).

Практичне значення одержаних результатів. Розроблено програмний комплекс мовою Python із використанням фреймворку TensorFlow/Keras, здатний в автоматичному режимі з точністю понад 91% диференціювати нормальні рентгенограми, прояви COVID-19, неспецифічні легеневі помутніння та вірусні пневмонії. Вбудований модуль генерації теплових карт дозволяє локалізувати вогнища патологій, що робить розроблену систему придатною для використання в якості асистента лікаря-рентгенолога (Computer-Aided Diagnosis).

У процесі підготовки бакалаврської кваліфікаційної роботи застосовувалися технології штучного інтелекту як допоміжний інструментарій. Зокрема, для стилістичної правки та структурування тексту використано ChatGPT-5 та Gemini 3.3. Автор несе повну відповідальність за зміст роботи

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ДАНИХ

1.1. Проблематика діагностики захворювань легень за допомогою рентгенографії

Рентгенографія органів грудної клітки є одним із найпоширеніших, найдоступніших та економічно найефективніших методів медичної візуалізації, що рутинно використовується для первинного скринінгу та діагностики респіраторних захворювань. Незважаючи на появу більш точних методів, таких як комп’ютерна томографія (КТ), рентгенографія залишається "золотим стандартом" першої лінії завдяки низькому променевому навантаженню на пацієнта та можливості проведення досліджень за допомогою портативних апаратів в умовах інтенсивної терапії.

Однак аналіз рентгенівських знімків супроводжується низкою суттєвих клінічних та технічних проблем. По-перше, проекційна природа рентгенограми призводить до накладання тривимірних анатомічних структур (ребер, серця, магістральних судин) на двовимірну площину знімка. Це створює інтенсивний структурний шум, який маскує мікроознаки патологічних процесів у паренхімі легень.

По-друге, ключовою проблемою є високий рівень візуальної подібності між різними типами легеневих патологій. Зокрема, інтерстиціальні зміни, спричинені вірусною пневмонією, та специфічні патерни коронавірусної інфекції (COVID-19) часто проявляються у вигляді двобічних периферичних помутнінь за типом "матового скла" та фокальних консолідацій [7]. Диференціація цих станів від неспецифічних легеневих помутнінь (Lung Opacity) вимагає високого рівня експертизи.

По-третє, суб'єктивний фактор відіграє критичну роль. В умовах високого навантаження на медичну систему (що особливо яскраво проявилося під час пандемії COVID-19) зростає ризик лікарських помилок через

когнітивну втому. Рівень інтер- та інтра-експертної варіабельності (розбіжності у діагнозах між різними лікарями або одним лікарем у різний час) при аналізі складних рентгенограм може досягати 20-30% [1].

Отже, існує гостра необхідність у розробці систем автоматизованої діагностики (Computer-Aided Diagnosis), здатних об'єктивно, швидко та з високою точністю класифікувати візуальні патерни легеневих захворювань, мінімізуючи вплив людського фактора та прискорюючи процес прийняття клінічних рішень.

1.2. Огляд існуючих методів комп'ютерного зору в медичній візуалізації

Еволюція методів комп'ютерного зору в галузі медичної візуалізації поділяється на два основні етапи: використання класичного машинного навчання із ручним виділенням ознак та перехід до методів глибокого навчання.

Класичні підходи базувалися на застосуванні фільтрів Габора, аналізі текстур (наприклад, матриці співрозмірності сірих тонів – GLCM) або локальних бінарних патернів (LBP) з подальшою класифікацією за допомогою машин опорних векторів (SVM) чи випадкових лісів (Random Forest). Фундаментальним недоліком таких методів є їхня вразливість до змін в умовах зйомки (освітленість, контраст, позиціонування пацієнта) та нездатність фіксувати складні, нелінійні ієрархічні ознаки [2].

Сучасним стандартом в аналізі медичних зображень є використання згорткових нейронних мереж (Convolutional Neural Networks, CNN). Завдяки операціям згортки та пулінгу, CNN здатні автоматично формувати ієрархію візуальних абстракцій: від простих країв та градієнтів на перших шарах до складних патологічних патернів на глибоких рівнях.

Важливим проривом у розвитку CNN стала поява архітектур із залишковими зв'язками (Residual Networks, ResNet), запропонованих K. He та

співавторами [5]. Введення skip-connections дозволило вирішити проблему згасання градієнта, що дало змогу тренувати надглибокі мережі (50, 101, 152 шари) з високою здатністю до генералізації візуальних ознак.

Через обмеженість обсягів анотованих медичних баз даних ключовим методом у сучасній радіономіці стало перенесення навчання (Transfer Learning) [3]. Використання моделей, попередньо навчених на великомасштабних загальних масивах даних (наприклад, ImageNet), дозволяє ініціалізувати ваги мережі оптимальними значеннями, після чого проводиться тонке налаштування на специфічному медичному датасеті. Це суттєво зменшує ризик перенавчання та прискорює збіжність алгоритму [6].

Окремим вектором розвитку є забезпечення інтерпретованості рішень штучного інтелекту (Explainable AI, XAI). Оскільки нейронні мережі є "чорними ящиками", клінічна валідація їхніх результатів потребує візуального пояснення. Найбільш ефективним методом вирішення цієї задачі є алгоритм Grad-CAM (Gradient-weighted Class Activation Mapping) [4], який генерує просторові теплові карти, відображаючи градієнти цільового класу на останньому згортковому шарі. Це дозволяє верифікувати, чи спирається модель на реальні патологічні ознаки у паренхімі легень, а не на артефакти зображення.

1.3. Аналіз доступних масивів рентгенівських зображень

Ефективність методів глибокого машинного навчання безпосередньо залежить від обсягу, якості та репрезентативності навчальних даних. У сфері радіологічної діагностики існує декілька відкритих масивів (наприклад, NIH ChestX-ray14, CheXpert), проте більшість із них сфокусована на класичних патологіях і не містить достатньої вибірки специфічних вірусних пневмоній, зокрема COVID-19.

У якості емпіричної бази даного дослідження обрано спеціалізований масив даних COVID-19 Radiography Database, розроблений групою

дослідників спільно з медичними фахівцями (Qatar University, University of Dhaka) [8]. Загальний обсяг очищеного датасету становить 21165 рентгенограм органів грудної клітки, що представлені у форматі PNG із просторовою роздільною здатністю 299×299 пікселів. Дані розподілені за чотирма діагностичними категоріями (рис. 1.1).

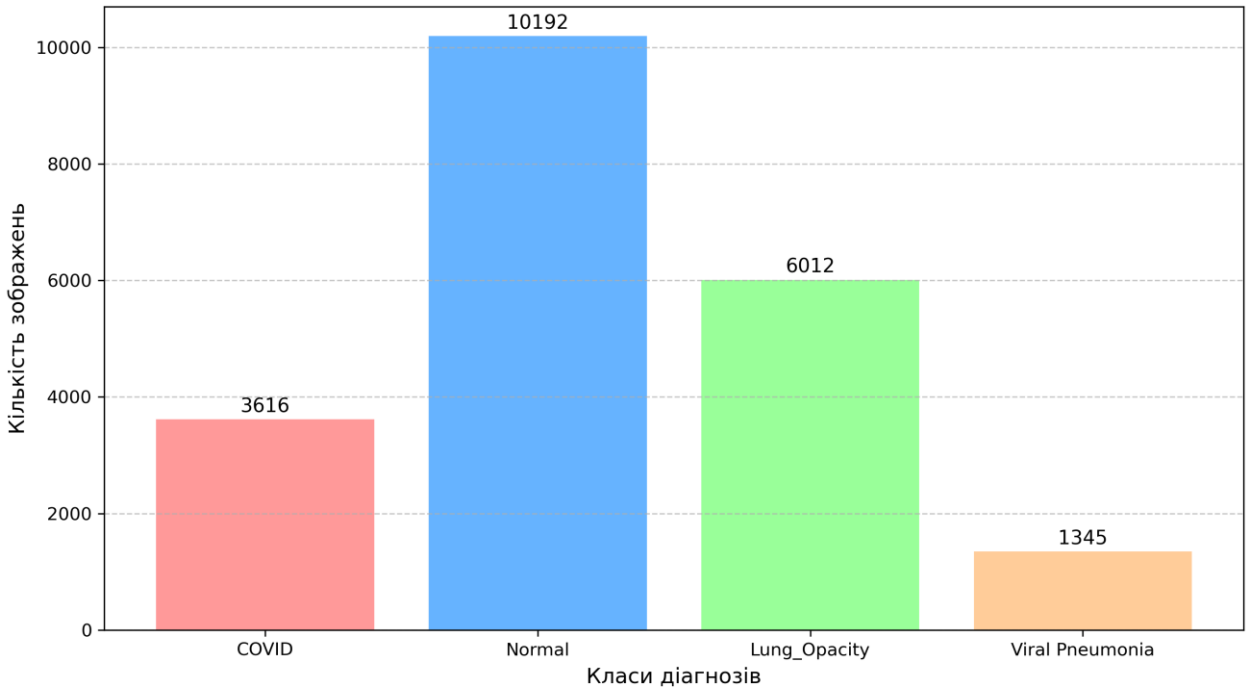


Рисунок 1.1 – Гістограма розподілу класів

Проведений під час підготовки даних розвідувальний аналіз виявив наступний кількісний розподіл спостережень:

- клас «Normal» (здорові пацієнти) налічує 10 192 зображення, що становить 48,15% від загальної вибірки;
- клас «Lung Opacity» (неспецифічні легеневі помутніння) – 6012 зображень (28,41%);
- клас «COVID» (патерни коронавірусної інфекції) – 3616 зображень (17,08%);
- клас «Viral Pneumonia» (класична вірусна пневмонія) – 1345 зображень (6,35%).

Отримана статистика свідчить про гетерогенність вибірки та наявність вираженого структурного дисбалансу класів [6]. Коефіцієнт дисбалансу

(Imbalance Ratio, IR) між мажоритарним класом («Normal») та міноритарним («Viral Pneumonia») становить приблизно 7,58.

Наявність такого дисбалансу є критичним фактором при проектуванні систем штучного інтелекту. Без застосування алгоритмічних компенсаторних механізмів (таких як зважування функції втрат або аугментація міноритарних класів), нейронна мережа буде схильна до зміщення прогнозів у бік мажоритарного класу, що є неприпустимим у медичній практиці через ризик хибнонегативних результатів діагностики.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи проведено системний аналіз предметної області, що дозволив сформулювати такі висновки:

1. Діагностика респіраторних захворювань за допомогою двовимірної рентгенографії ускладнюється структурним шумом (накладанням анатомічних утворень), високою візуальною подібністю різних типів патологій (COVID-19, вірусна пневмонія) та впливом суб'єктивного людського фактора. Це зумовлює об'єктивну необхідність розробки автоматизованих систем підтримки прийняття клінічних рішень.

2. Огляд існуючих методів комп'ютерного зору довів, що найбільш ефективним математичним апаратом для класифікації медичних зображень є глибокі згорткові нейронні мережі. Для компенсації обмеженого обсягу медичних баз даних доцільним є використання парадигми Transfer Learning із застосуванням попередньо навчених моделей (зокрема, архітектур сімейства ResNet) у поєднанні з методами візуальної інтерпретованості (Grad-CAM).

3. Аналіз обраної емпіричної бази (COVID-19 Radiography Database, 21165 зображень) підтвердив достатність масиву для навчання глибоких нейромереж. Водночас виявлено суттєвий міжкласовий дисбаланс (IR = 7,58) з домінуванням класу «Normal». Ця особливість даних вимагає адаптації стандартного процесу оптимізації моделі, використання зваженої функції

втрат та вибору специфічних метрик оцінювання якості класифікації, нечутливих до асиметрії вибірки.

Сформульовані висновки є базисом для подальшої математичної постановки задачі та вибору методів її розв'язання, що розглядаються у наступному розділі.

РОЗДІЛ 2
ПОСТАНОВКА ЗАДАЧІ ТА ВИБІР МЕТОДІВ РОЗВ'ЯЗАННЯ

2.1. Математична постановка задачі класифікації медичних зображень

З точки зору машинного навчання, діагностика легеневих патологій за рентгенівськими знімками розглядається як задача мультикласової класифікації з одним етикетуванням (single-label multi-class classification).

Нехай задано простір вхідних ознак $X \subset \mathbb{R}^{H \times W \times C}$, де кожне спостереження $x_i \in X$ є тензором цифрового зображення висоти *H*, ширини *W* та кількістю колірних каналів *C*. У нашому випадку після попередньої обробки H = 224, W = 224, C = 3. Нехай Y – множина можливих класів діагнозів, де |Y| = K = 4, а саме: $Y = \{COVID, Normal, Lung_Opacity, Viral_Pneumonia\}$. Кожному зображенню x_i відповідає мітка $y_i \in \{0,1\}^K$, представлена у вигляді унітарного коду (one-hot encoding), де $\sum_{k=1}^{K} y_{ik} = 1$.

Метою дослідження є знаходження оптимального відображення $f : X \to \mathcal{P}(Y)$, де $\mathcal{P}(Y)$ – простір розподілів ймовірностей над множиною класів. Відображення апроксимується параметричною моделлю (згортковою нейронною мережею) $f(x, \theta)$, де θ – вектор ваг мережі. Вихідним сигналом моделі є вектор ймовірностей $\hat{y}_i$, отриманий за допомогою функції активації Softmax на останньому шарі:

$$\hat{y}_{ik} = P(y_i = k \mid x_i; \theta) = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)}$$

де z_k – вихід лінійного перетворення (логіт) для класу k.

Для знаходження оптимальних параметрів θ^* необхідно мінімізувати емпіричний ризик на навчальній вибірці $D = \{(x_i, y_i)\}_{i=1}^{N}$. Враховуючи

виявлений раніше дисбаланс класів, використання стандартної багатокласової перехресної ентропії (Categorical Cross-Entropy) призведе до зміщення градієнтів на користь мажоритарного класу. Тому математична постановка оптимізаційної задачі базується на мінімізації зваженої функції втрат (Weighted Categorical Cross-Entropy):

$$\theta^* = \arg\min_\theta \left(-\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} w_k \cdot y_{ik} \cdot \log(\hat{y}_{ik}) \right)$$

де w_k – ваговий пеналізаційний коефіцієнт k-го класу. Для забезпечення балансування вагові коефіцієнти обчислюються обернено пропорційно частотам класів у навчальній вибірці:

$$w_k = \frac{N}{K \cdot N_k}$$

де N – загальна кількість спостережень, N_k – кількість спостережень у k-му класі. Такий підхід гарантує, що помилка класифікації міноритарного класу (наприклад, вірусної пневмонії) генеруватиме еквівалентний за модулем градієнт порівняно з мажоритарним класом (норма).

2.2. Вибір метрик для оцінки ефективності моделей в умовах дисбалансу класів

Оцінка якості класифікаційних моделей у медичній діагностиці вимагає специфічного підходу. З огляду на структурний дисбаланс масиву даних (де 48,15% вибірки належить до класу «Normal»), використання базової метрики загальної точності (Accuracy) є математично некоректним та призводить до "парадоксу точності". Наївна модель, що класифікуватиме всі спостереження як норму, отримає Accuracy на рівні ~48%, маючи при цьому нульову предиктивну здатність щодо патологій.

Відповідно, для об'єктивної оцінки ефективності алгоритмів використовується матриця похибок та похідні від неї метрики: істинно

позитивні (TP), хибнопозитивні (FP), істинно негативні (TN) та хибнонегативні (FN) рішення.

У контексті медичного скринінгу найбільш критичною метрикою є повнота або чутливість (Recall / Sensitivity), оскільки вартість хибнонегативного прогнозу (пропуск хворої людини) значно перевищує вартість хибнопозитивного (призначення додаткового обстеження здоровому пацієнту)

$$Recall_k = \frac{TP_k}{TP_k + FN_k}.$$

Точність класифікації (Precision) визначає частку дійсно хворих пацієнтів серед усіх, кого модель віднесла до певної патології:

$$Precision_k = \frac{TP_k}{TP_k + FP_k}$$

Для отримання єдиного збалансованого критерію якості оптимізації, який враховує як Precision, так і Recall, використовується гармонійне середнє – F1-міра:

$$F1_k = 2 \cdot \frac{Precision_k \cdot Recall_k}{Precision_k + Recall_k}$$

Оскільки задача є мультикласовою, а дані – незбалансованими, глобальна оцінка моделі здійснюється за допомогою макро-усереднення. На відміну від мікро-усереднення, яке агрегує загальну кількість TP, FP та FN по всій вибірці, макро-усереднення обчислює метрики незалежно для кожного класу, після чого знаходить їхнє арифметичне середнє:

$$Macro\ F1 = \frac{1}{K}\sum_{k=1}^{K} F1_k.$$

Ця метрика надає однакову вагу кожному класу незалежно від його розміру, що робить її найдоцільнішим критерієм ефективності розроблюваної системи автоматичної класифікації рентгенівських зображень.

2.3. Обґрунтування вибору архітектур згорткових нейронних мереж

Аналіз медичних зображень вимагає алгоритмів, здатних враховувати просторову топологію пікселів. Стандартні багатошарові персептрони не підходять для цієї задачі, оскільки вони руйнують двовимірну структуру зображення та потребують надмірної кількості параметрів. Оптимальним інструментом для виділення візуальних ознак є згорткові нейронні мережі.

Основна перевага CNN полягає у використанні локальних рецептивних полів (операція згортки) та спільному використанні ваг. Мережа послідовно агрегує інформацію: перші шари реагують на базові ознаки (краї, градієнти тіней), тоді як глибші шари формують абстрактні уявлення складних структур, таких як вогнища консолідації у легенях або патерни "матового скла".

Однак, як показує практика, спроба навчити базову згорткову мережу "з нуля" (з випадковою ініціалізацією ваг) на відносно невеликих медичних датасетах часто призводить до явища перенавчання та нестабільності градієнтів. Збільшення глибини мережі для підвищення її виразної здатності спричиняє проблему згасання градієнта, через що навчання стає неможливим.

Для подолання цих фундаментальних обмежень у даній роботі обрано архітектуру ResNet (Residual Network), зокрема модифікацію ResNet50. Головною алгоритмічною інновацією цієї моделі є використання залишкових блоків із так званими "skip connections" (обхідними зв'язками). Вони дозволяють передавати вхідний сигнал безпосередньо через кілька шарів вперед, що забезпечує безперешкодне поширення градієнта навіть у дуже глибоких мережах.

Крім того, для компенсації дефіциту розміченої медичної вибірки застосовується парадигма перенесення навчання. Замість ініціалізації випадковими значеннями, базові шари ResNet50 завантажуються з вагами, попередньо навченими на масиві ImageNet (який містить понад мільйон зображень). Це дозволяє мережі відразу ефективно розпізнавати базові візуальні патерни. Під час подальшого навчання на рентгенограмах

відбувається лише точне налаштування останніх класифікаційних шарів моделі, що кардинально підвищує стабільність навчання та загальну точність діагностики.

2.4. Методи підвищення інтерпретованості рішень штучного інтелекту в медицині

Незважаючи на високу предиктивну ефективність глибоких нейронних мереж, їхнім суттєвим недоліком у клінічній практиці є проблема "чорного ящика". Система генерує діагноз (вихідні ймовірності), але не пояснює логіку свого рішення. Для впровадження алгоритмів штучного інтелекту в медичний процес критично важливо забезпечити їхню прозорість, щоб лікар-рентгенолог міг верифікувати результати. З цією метою у роботі застосовуються методи Explainable AI (XAI).

Найбільш адаптивним підходом для архітектур CNN є алгоритм Grad-CAM (Gradient-weighted Class Activation Mapping). Його концептуальна суть полягає у візуалізації тих регіонів зображення, які мали найбільший вплив на формування підсумкової оцінки для конкретного класу патології.

Алгоритм Grad-CAM працює з останнім згортковим шаром мережі, оскільки саме він містить найвищий рівень просторово-семантичної інформації. Процедура побудови теплової карти включає наступні кроки:

1. Для обраного класу (наприклад, COVID-19) обчислюється градієнт вихідного сигналу моделі відносно карт ознак останнього згорткового шару. Це дозволяє визначити, наскільки кожна карта ознак важлива для розпізнавання даної патології.

2. Проводиться глобальне усереднення цих градієнтів, що дає набір скалярних вагових коефіцієнтів для кожної карти ознак (визначається "важливість" конкретного фільтра).

3. Виконується зважена сума карт ознак останнього шару з використанням знайдених коефіцієнтів.

4. До отриманого результату застосовується операція ReLU, щоб залишити лише ті пікселі, які здійснюють позитивний вплив на цільовий клас (відсікаються регіони, що належать до інших діагнозів).

Отримана теплова карта масштабується до розмірів оригінальної рентгенограми та накладається на неї. В результаті формується зображення, де інтенсивність кольору (від синього до червоного) вказує на зони максимальної уваги моделі. Якщо червоні зони корелюють із реальним розташуванням легеневої паренхіми та вогнищами уражень, рішення моделі вважається науково та клінічно обґрунтованим.

Висновки до розділу 2

У другому розділі здійснено математичне та алгоритмічне обґрунтування методів класифікації медичних зображень:

1. Задачу діагностики формалізовано як мінімізацію зваженої функції втрат багатокласової перехресної ентропії. Використання вагових коефіцієнтів для різних діагнозів дозволяє нівелювати вплив апріорного міжкласового дисбалансу вибірки.

2. Обґрунтовано вибір метрик оцінювання якості. Доведено недоцільність використання метрики Accuracy на користь макро-усередненої F1-міри (Macro-F1) та метрики повноти (Recall), що мінімізує ризик хибнонегативних прогнозів, критичних для медичної сфери.

3. Доведено доцільність застосування архітектури ResNet50 у комбінації з парадигмою Transfer Learning. Використання залишкових зв'язків та попередньо навчених ваг вирішує проблему згасання градієнта та забезпечує стабільну генералізацію на обмежених обсягах медичних даних.

4. Обрано алгоритм Grad-CAM як базовий механізм інтерпретації результатів. Генерація просторових теплових карт дозволяє подолати проблему "чорного ящика" та забезпечує клінічну верифікацію роботи моделі на основі локалізації легеневих патологій.

РОЗДІЛ 3
РОЗРОБКА ТА ІМПЛЕМЕНТАЦІЯ РІШЕННЯ

3.1. Загальна архітектура системи автоматичної класифікації зображень

Розробка системи автоматизованої діагностики захворювань легень реалізована у вигляді багатокрокового конвеєра обробки даних (data pipeline). Програмна реалізація виконана мовою Python з використанням відкритої бібліотеки машинного навчання TensorFlow та високорівневого API Keras. Загальна архітектура системи складається з чотирьох логічних модулів:

1. **Модуль завантаження та фільтрації даних.** На цьому етапі алгоритм сканує ієрархію директорій обраного масиву COVID-19 Radiography Database. Оскільки вихідний датасет містить додаткові допоміжні файли (маски сегментації легень), розроблено скрипт на базі бібліотеки Pandas, який алгоритмічно відфільтровує оригінальні рентгенограми, виключаючи потрапляння масок до навчальної вибірки. Створюється структурований фрейм даних, що містить абсолютні шляхи до файлів та відповідні їм мітки класів.

2. **Модуль попередньої обробки (Preprocessing Module).** Відповідає за приведення гетерогенних візуальних даних до єдиного тензорного формату, стратифікований розподіл вибірки та динамічну аугментацію в оперативній пам'яті під час навчання.

3. **Модуль нейромережевої класифікації.** Включає в себе архітектури моделей (базову згорткову мережу та адаптовану ResNet50). Модуль ініціалізує ваги, визначає топологію шарів, функцію втрат, оптимізатор (Adam) та застосовує механізми регуляризації (Early Stopping, Dropout) для управління процесом мінімізації помилки.

4. **Модуль валідації та інтерпретації.** Відповідає за обчислення експлуатаційних метрик (Loss, Accuracy, Precision, Recall) на кожній епосі

навчання, збереження найкращих конфігурацій ваг (ModelCheckpoint) та генерацію просторових теплових карт за допомогою алгоритму Grad-CAM для забезпечення прозорості прийнятих системою рішень.

Такий модульний підхід забезпечує масштабованість системи та дозволяє незалежно модифікувати окремі етапи конвеєра (наприклад, замінити архітектуру мережі) без втручання в алгоритми обробки даних.

3.2. Попередня обробка та аугментація масивів візуальних даних

Підготовка вхідних даних є критичним етапом, що безпосередньо впливає на збіжність алгоритмів оптимізації. Процес попередньої обробки імплементовано за допомогою класу ImageDataGenerator.

Оскільки вхідні рентгенограми мають різний початковий розмір, на першому кроці всі зображення алгоритмічно масштабуються до розмірності 224x224x3 пікселів за допомогою білінійної інтерполяції. Це стандартний розмір вхідного тензора для сучасних архітектур комп'ютерного зору (зокрема ResNet). Для базової моделі значення пікселів нормалізуються з діапазону [0, 255] до відрізка [0, 1]. Для моделі на базі ResNet застосовується специфічна функція preprocess_input, яка виконує центрування значень пікселів відносно середнього показника каналів масиву ImageNet.

Загальний масив даних (21165 зображень) було розділено на навчальну та валідаційну вибірки у пропорції 80/20 (16932 та 4233 зображень відповідно). Розбиття здійснено із застосуванням методу стратифікації, що гарантує збереження однакового співвідношення класів у обох підвибірках.

З метою підвищення стійкості моделі до просторових трансформацій та запобігання перенавчанню, до навчальної вибірки застосовано метод динамічної аугментації даних. Згенеровано наступний набір трансформацій:

- випадковий поворот зображення (rotation_range) в межах $\pm 15^{\circ}$;
- горизонтальний та вертикальний зсуви (width/height_shift_range) на 10% від розміру зображення;

- випадкове масштабування (zoom_range) у межах 10%.

Важливе клінічне обмеження: в процесі аугментації було свідомо відключено функцію випадкового горизонтального віддзеркалення. Анатомічно серце людини розташоване зліва, і його тінь є ключовим орієнтиром на рентгенограмі. Горизонтальне віддзеркалення створило б біологічно некоректні зображення (штучну декстрокардію), що дезорієнтувало б нейронну мережу під час виділення просторових ознак легеневих полів. Валідаційна вибірка не піддавалася аугментації для забезпечення чистоти експерименту.

Для вирішення проблеми виявленого міжкласового дисбалансу (описаного у підрозділі 1.3) застосовано метод зважування класів. Відповідно до частотного розподілу навчальної вибірки, за допомогою алгоритму compute_class_weight розраховано пеналізаційні коефіцієнти для зваженої функції втрат. Мажоритарний клас «Normal» отримав найменшу вагу $w \approx 0,52$, міноритарний клас «Viral Pneumonia» – максимальну $w \approx 3,93$, класи «COVID» та «Lung Opacity» отримали коефіцієнти 1,46 та 0,88 відповідно. Це забезпечило збалансоване оновлення ваг градієнтним спуском незалежно від частоти появи класу у батчі.

Для програмної імплементації методу зважування класів використано функціонал бібліотеки scikit-learn. Алгоритм автоматично обчислює обернені частоти для балансування функції втрат (Лістинг 3.1).

Лістинг 3.1. Фрагмент коду балансування функції втрат

```
from sklearn.utils.class_weight import compute_class_weight
# Розрахунок вагових коефіцієнтів класів
class_weights_array = compute_class_weight(
    class_weight='balanced',
    classes=np.unique(train_gen.classes),
    y=train_gen.classes)
class_weights_dict = dict(enumerate(class_weights_array))
```

Кінець лістингу 3.1

3.3. Розробка базової моделі згорткової нейронної мережі

Для встановлення базового рівня продуктивності та підтвердження гіпотези про необхідність використання глибоких архітектур, на першому етапі було спроектовано класичну згорткову нейронну мережу (CNN). Модель розроблено з використанням класу Sequential бібліотеки Keras, що передбачає лінійний стек шарів.

Архітектура розробленої базової CNN складається з трьох каскадних блоків виділення ознак (Feature Extractors) та одного класифікаційного блоку. Кожен з трьох згорткових блоків містить:

1. Шар двовимірної згортки Conv2D із розміром ядра 3x3 та функцією активації ReLU (Rectified Linear Unit), яка вводить нелінійність у модель. Кількість фільтрів послідовно збільшується: 32, 64 та 128 для першого, другого та третього блоків відповідно. Це дозволяє мережі на кожному рівні формувати все більш складні візуальні абстракції.

2. Шар пакетної нормалізації BatchNormalization, який центрує та масштабує активації попереднього шару. Це стабілізує розподіл входів, нівелює проблему внутрішнього коваріативного зсуву (Internal Covariate Shift) та прискорює збіжність алгоритму оптимізації.

3. Шар субдискретизації MaxPooling2D (2x2), що виконує просторове стиснення карт ознак, зменшуючи обчислювальну складність та забезпечуючи локальну інваріантність до трансляцій.

Після етапу вилучення ознак багатовимірний тензор перетворюється на одновимірний вектор за допомогою шару Flatten. Класифікаційна частина складається з повнозв'язного шару Dense на 128 нейронів. Для запобігання перенавчанню застосовано метод регуляризації Dropout із коефіцієнтом відключення нейронів $p = 0.5$. Вихідний шар містить 4 нейрони (відповідно до кількості діагностичних класів) із функцією активації Softmax, що генерує дискретний розподіл ймовірностей.

Детальна топологія розробленої базової згорткової нейронної мережі, зміна розмірностей просторових тензорів на кожному етапі перетворення та розподіл оптимізаційних параметрів (ваг) наведені у Таблиці 3.1.

Таблиця 3.1 – Архітектура та параметри базової згорткової моделі

Назва шару (Тип)	Розмірність вихідного тензора	Кількість параметрів	Призначення
Вхідний тензор	(224, 224, 3)	0	Нормалізоване RGB-зображення
Conv2D_1 (Згортка 3×3)	(222, 222, 32)	896	Виділення низькорівневих ознак
BatchNormalization_1	(222, 222, 32)	128	Центрування активацій
MaxPooling2D_1 (2×2)	(111, 111, 32)	0	Просторове стиснення
Conv2D_2 (Згортка 3×3)	(109, 109, 64)	18 496	Виділення ознак середнього рівня
BatchNormalization_2	(109, 109, 64)	256	Центрування активацій
MaxPooling2D_2 (2×2)	(54, 54, 64)	0	Просторове стиснення
Conv2D_3 (Згортка 3×3)	(52, 52, 128)	73 856	Формування складних патернів
BatchNormalization_3	(52, 52, 128)	512	Центрування активацій
MaxPooling2D_3 (2×2)	(26, 26, 128)	0	Просторове стиснення
Flatten (Векторизація)	(86528)	0	Перехід до одновимірного простору
Dense_1 (Повнозв'язний)	(128)	11 075 712	Класифікаційний шар
Dropout (p = 0.5)	(128)	0	Просторова регуляризація
Dense_2 (Вихідний)	(4)	516	Розподіл ймовірностей (Softmax)

Загальна кількість параметрів моделі: 11170372 (усі параметри доступні для навчання).

Навчання моделі здійснювалося за допомогою адаптивного оптимізатора Adam (Adaptive Moment Estimation) з початковою швидкістю

навчання (learning rate) $\eta = 0.001$. Для автоматизованого керування процесом оптимізації імплементовано механізм ReduceLROnPlateau, який експоненційно зменшує швидкість навчання вдвічі при стагнації функції втрат, та EarlyStopping (рання зупинка) для припинення навчання у разі ознак перенавчання.

3.4. Програмна реалізація методів перенесення навчання

Зважаючи на високу дисперсію базових CNN при роботі зі складними медичними зображеннями, основним методом вирішення задачі обрано парадигму перенесення навчання. У якості базової архітектури використано глибоку нейронну мережу ResNet50, яка містить 50 шарів із залишковими зв'язками.

Процес ініціалізації попередньо навченої моделі та реалізації механізму Feature Extraction (заморожування ваг згорткової бази) наведено у Лістингу 3.2.

Лістинг 3.2. Фрагмент ініціалізації архітектури Transfer Learning

```
# Завантаження базової моделі ResNet50 без верхнього класифікатора
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
# Заморожування ваг базової моделі (Feature Extraction)
base_model.trainable = False
model = Sequential([
    base_model,
    GlobalAveragePooling2D(),
    BatchNormalization(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax')
])
```

Кінець лістингу 3.2

Програмна реалізація виконана за стратегією виділення ознак. Екземпляр моделі ResNet50 завантажено за допомогою модуля tf.keras.applications без верхнього повнозв'язного класифікатора (include_top=False) та з вагами, попередньо навченими на великомасштабному масиві загального призначення ImageNet. Усі згорткові шари базової моделі були переведені у нетренований стан (base_model.trainable = False), тобто їхні ваги були "заморожені" і не оновлювалися під час алгоритму зворотного поширення помилки. Це дозволило зберегти вже сформовану здатність мережі розпізнавати базові візуальні структури (краї, текстури) та суттєво зменшити обчислювальні витрати.

Поверх замороженої згорткової бази було сконструйовано новий класифікаційний блок, оптимізований під задачу медичної діагностики:

1. Шар GlobalAveragePooling2D. На відміну від операції Flatten, глобальний пулінг усереднює кожну карту ознак до єдиного скалярного значення. Це радикально зменшує загальну кількість параметрів моделі (запобігаючи перенавчанню) та робить архітектуру інваріантною до просторових зсувів вогнищ патологій на рентгенограмі.

2. Шар пакетної нормалізації BatchNormalization.

3. Повнозв'язний прихований шар Dense на 256 нейронів (активація ReLU).

4. Шар просторової регуляризації Dropout (p = 0.5).

5. Вихідний шар Dense на 4 класи (активація Softmax).

Структура адаптованої нейронної мережі на базі архітектури ResNet50 із застосуванням парадигми Transfer Learning наведена у Таблиці 3.2. Особливістю даної архітектури є те, що понад 23 мільйони параметрів базової моделі виключені з процесу градієнтного спуску («заморожені»), що радикально знижує обчислювальне навантаження та ризик перенавчання.

Таблиця 3.2 – Архітектура моделі на базі Transfer Learning (ResNet50)

Назва шару (Тип)	Розмірність вихідного тензора	Кількість параметрів	Стан для навчання
Вхідний тензор	(224, 224, 3)	0	-
ResNet50 (Базова мережа)	(7, 7, 2048)	23 587 712	Заморожено (Non-trainable)
GlobalAveragePooling2D	(2048)	0	-
BatchNormalization	(2048)	8 192	Тренується (Trainable)
Dense_1 (Повнозв'язний)	(256)	524 544	Тренується (Trainable)
Dropout (p = 0.5)	(256)	0	-
Dense_2 (Вихідний)	(4)	1 028	Тренується (Trainable)

Загальна кількість параметрів: 24121476 Кількість параметрів, що навчаються: 529668 (лише класифікаційний блок) Кількість заморожених параметрів: 23591808

Компіляція та конфігурація колбеків (callbacks) даної моделі були ідентичні базовій CNN, що забезпечило методологічну чистоту експерименту та об'єктивність подальшого порівняльного аналізу ефективності обох підходів.

Висновки до розділу 3

У третьому розділі проведено практичну імплементацію системи класифікації медичних зображень.

1. Розроблено гнучкий багатокроковий конвеєр для завантаження, фільтрації та обробки масивів рентгенограм на базі фреймворку TensorFlow/Keras.

2. Реалізовано алгоритми попередньої підготовки даних. Застосування динамічної просторової аугментації (без горизонтального віддзеркалення для збереження біологічної коректності) та розрахунок зважувальних коефіцієнтів класів дозволили мінімізувати вплив структурного дисбалансу датасету на процес навчання мережі.

3. Спроектовано та запрограмовано дві нейромережеві архітектури: базову згорткову мережу, що навчалася з випадковою ініціалізацією ваг, та модель на базі ResNet50 із використанням методу Transfer Learning ("заморожування" згорткової бази). Налаштування механізмів регуляризації (Dropout, Early Stopping) гарантувало коректний контроль процесу мінімізації емпіричного ризику в обох моделях.

РОЗДІЛ 4
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1. Методика проведення експериментів та налаштування гіперпараметрів

Експериментальне дослідження розроблених нейромережевих архітектур проводилося в єдиному програмно-апаратному середовищі для забезпечення об'єктивності порівняльного аналізу. Навчання та тестування моделей виконувалося на базі фреймворку глибокого навчання TensorFlow.

Для процесу оптимізації ваг було зафіксовано наступний набір гіперпараметрів. Розмірність пакету даних встановлено на рівні 32 зображення, що є оптимальним компромісом між швидкістю збіжності градієнтного спуску та обсягом споживаної відеопам'яті (VRAM). Максимальна кількість епох навчання (epochs) дорівнювала 15. Просторовий розмір вхідних тензорів становив 224x224 пікселів.

Мінімізація функції втрат (зваженої перехресної ентропії) здійснювалася за допомогою алгоритму стохастичної оптимізації Adam. Початкову швидкість навчання було встановлено на рівні $\eta = 0.001$. Оскільки медичні зображення характеризуються високим рівнем візуального шуму, використання статичного кроку навчання може призвести до осциляцій навколо глобального мінімуму. Тому було застосовано стратегію динамічного адаптивного керування кроком: алгоритм ReduceLROnPlateau зменшував швидкість навчання у 2 рази (factor = 0.5), якщо значення валідаційної функції втрат не покращувалося протягом 2 епох поспіль.

Оцінка генералізуючої здатності алгоритмів (здатності працювати з новими даними) проводилася виключно на валідаційній вибірці (20% від датасету, 4233 зображення), яка не брала участі в оновленні ваг моделі та не піддавалася просторовій аугментації. Для запобігання перенавчанню

використовувався механізм ранньої зупинки EarlyStopping (з параметром patience = 3-4), який автоматично переривав тренування при початку деградації валідаційних метрик та відновлював найкращу конфігурацію варіаційних ваг (restore_best_weights).

4.2. Аналіз результатів роботи базової моделі класифікації

Перший етап експериментального дослідження полягав у тестуванні розробленої базової архітектури згорткової нейронної мережі, що ініціалізувалася випадковими вагами. Метою даного етапу було встановлення емпіричного базового рівня продуктивності системи.

Процес оптимізації базової моделі продемонстрував її обмежену здатність до стабільної генералізації на складному гетерогенному масиві медичних даних. Динаміка навчання (див. рис. 4.1) характеризувалася високою дисперсією метрик: на графіках валідаційної вибірки спостерігалися різкі осциляції та відсутність плавного спадання функції втрат.

Під час навчання спрацював алгоритм ранньої зупинки, який перервав процес після 15 епох через відсутність подальшої мінімізації емпіричного ризику. Найкращі показники були зафіксовані на 13-й епосі, де функція втрат на валідаційній вибірці склала $Loss_{val} = 0.6779$. На цій ітерації модель досягла загальної точності (Accuracy) на рівні 77,51% та точності виявлення (Precision) 88,56%.

Проте макро-усереднена повнота (Recall), яка є найбільш критичною метрикою у медичній діагностиці, склала лише 60,69%. Це означає, що базова архітектура пропускала значну частку патологій, генеруючи хибнонегативні результати. Більше того, значна волатильність процесу на останніх епохах (зокрема, різке падіння валідаційної точності до 49,66% та зростання втрат до $Loss_{val} = 1.3006$ на 15-й епосі) є прямим свідченням нестабільності процесу оптимізації. Градієнти мережі, зіштовхуючись із високою візуальною

подібністю макроознак легень (ребер, контурів серця) різних класів, не змогли знайти стійкий локальний оптимум.

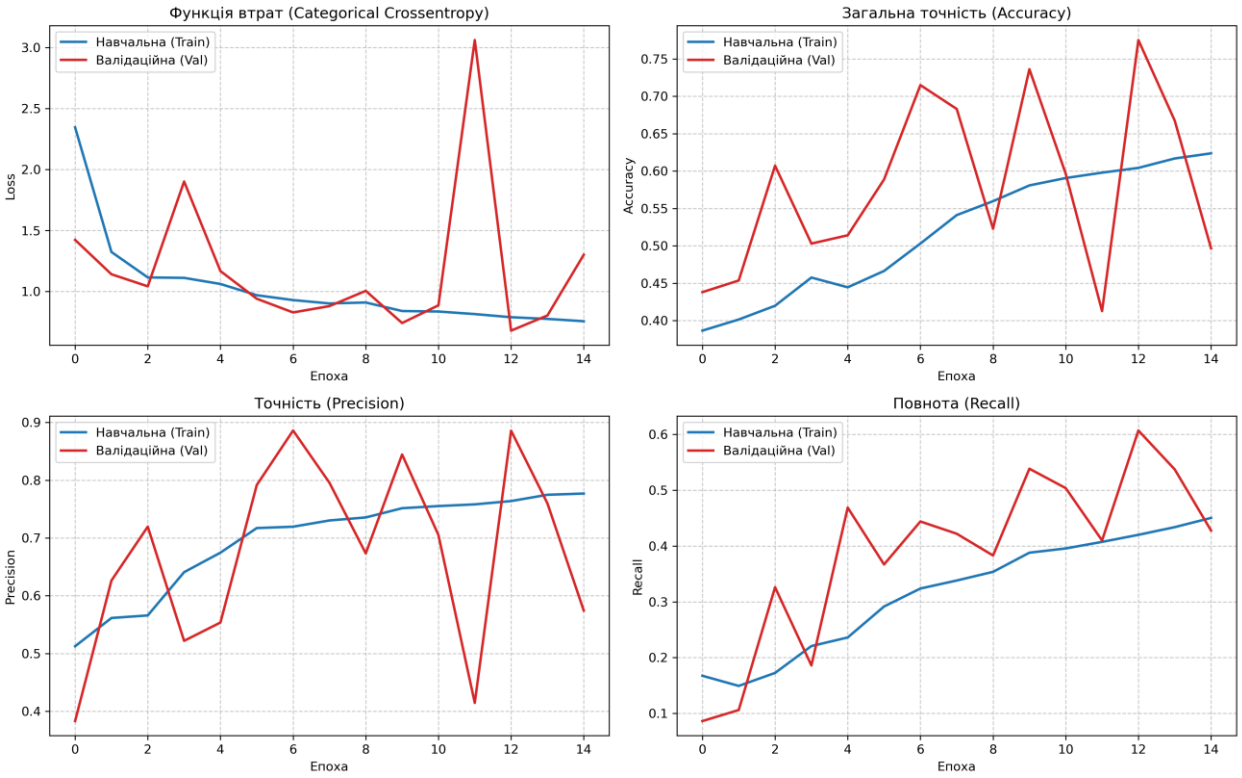


Рисунок 4.1 – Динаміка метрик навчання базової згорткової нейронної мережі

Отже, експериментальні результати довели, що класична згорткова архітектура, незважаючи на застосування методів регуляризації (Dropout, Batch Normalization) та балансування класів, є схильною до перенавчання та застрягання в локальних мінімумах. Ці емпіричні дані безпосередньо обґрунтовують необхідність переходу до парадигми перенесення навчання (Transfer Learning) та застосування глибоких попередньо навчених моделей для підвищення стабільності та діагностичної точності системи.

4.3. Аналіз результатів застосування попередньо навчених моделей

Для подолання проблеми високої дисперсії та низької узагальнюючої здатності базової CNN, на другому етапі експерименту було досліджено

ефективність архітектури на основі парадигми перенесення навчання із використанням попередньо навченої моделі ResNet50.

Експериментальні результати засвідчили значну перевагу даного підходу (див. рис. 4.2). Процес оптимізації відзначався високою алгоритмічною стабільністю: графіки функції втрат та метрик якості для навчальної і валідаційної вибірок демонстрували плавне, монотонне збігання без різких осциляцій. Це підтверджує, що ініціалізація ваг значеннями з масиву ImageNet дозволила моделі успішно оминути локальні мінімуми.

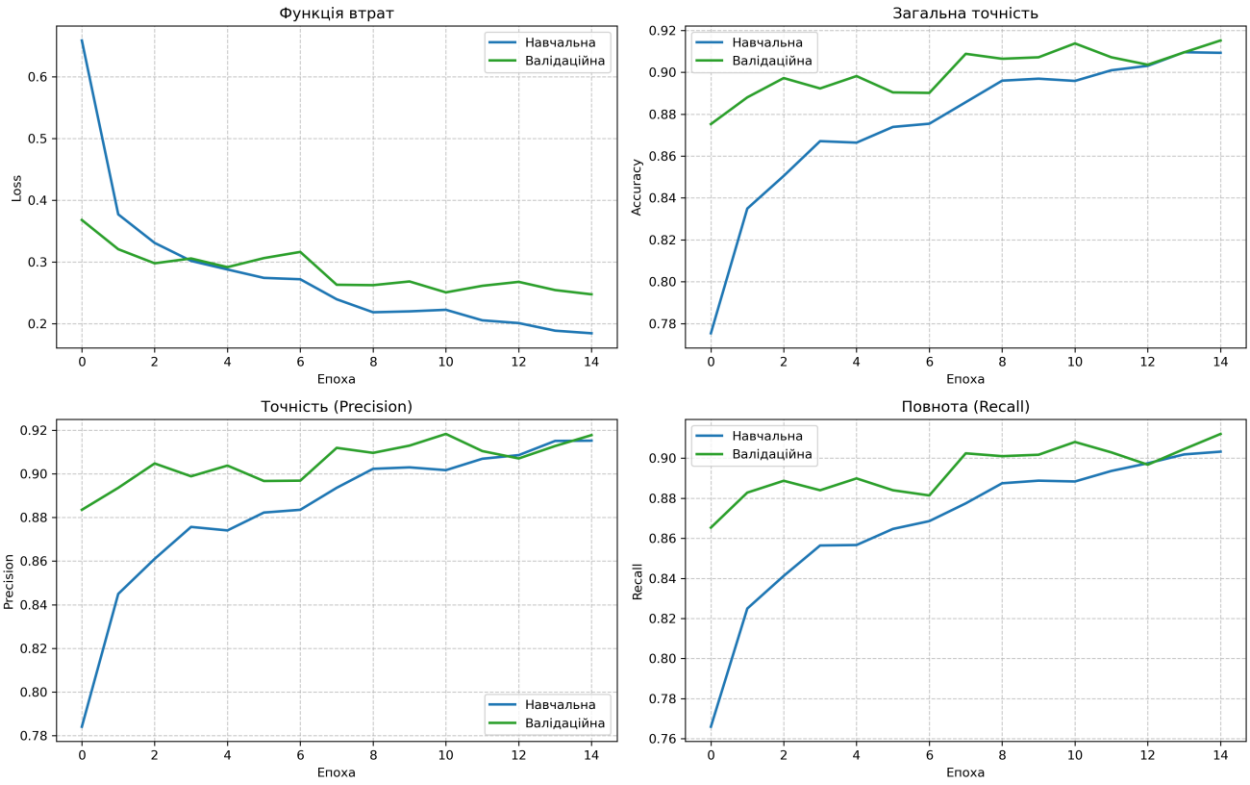


Рисунок 4.2 – Динаміка навчання моделі на базі архітектури ResNet50

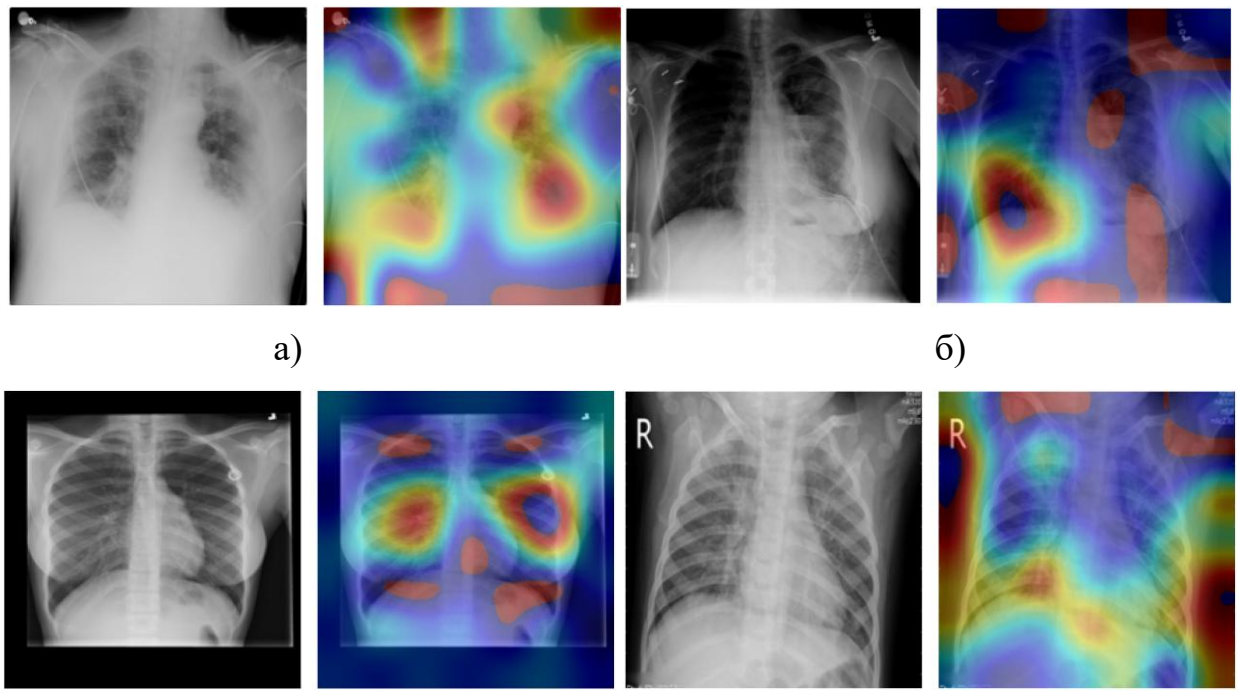
На 15-й епосі навчання модель досягла локального мінімуму функції втрат на валідаційній вибірці ($Loss_{val}$ = 0.2477). При цьому показник загальної точності (Accuracy) склав 91,52%, а точність виявлення (Precision) – 91,78%.

Особливо значущим здобутком для розроблюваної медичної системи є підвищення макро-усередненої повноти (Recall) до 91,21% (порівняно з 60,69% у базової моделі). Такий високий рівень чутливості вказує на те, що алгоритм коректно ідентифікує переважну більшість патологічних змін,

мінімізуючи кількість хибнонегативних прогнозів. Застосування механізму вилучення ознак із "замороженими" згортковими шарами запобігло перенавчанню моделі, незважаючи на об'єктивний структурний дисбаланс класів у навчальному масиві даних.

4.4. Візуалізація та інтерпретація ознак за допомогою методу Grad-CAM

Для забезпечення клінічної прозорості системи та верифікації того, що нейронна мережа приймає рішення на основі релевантних анатомічних ознак (а не графічних артефактів чи медичного маркування на знімках), було проведено просторову інтерпретацію вихідних даних за допомогою алгоритму Grad-CAM. Отримані теплові карти відображають зони максимальної активації останнього згорткового шару мережі для кожного з чотирьох класів діагнозів (див. рис. 4.3).



а) б)

в) г)

Рисунок 4.3 – Візуалізація Grad-CAM для класів:
а) COVID-19, б)Lung_Opacity, в) Normal, г) Viral Pneumonia

Аналіз згенерованих теплових карт дозволив зробити наступні клініко-топологічні висновки:

1. Клас COVID: зони найвищої активації (червоний спектр) локалізуються переважно на периферії обох легеневих полів. З медичної точки зору це є точним відображенням реального патогенезу: коронавірусна інфекція найчастіше проявляється у вигляді двобічних субплевральних помутнінь за типом "матового скла".

2. Клас Lung Opacity: модель безпомилково сфокусувала увагу на масивній асиметричній ділянці консолідації (щільному помутнінні) у правій легені, ігноруючи при цьому здорову тканину протилежної легені.

3. Клас Viral Pneumonia: теплова карта демонструє дифузний (розкиданий) патерн активації по всій площі легеневої паренхіми, що корелює з класичною рентгенологічною картиною інтерстиціального набряку при вірусних пневмоніях нековідної етіології.

4. Клас Normal: при аналізі знімка здорового пацієнта зони максимальної уваги мережі розташувалися поза межами внутрішньої паренхіми легень (на контурах ребер, діафрагми та ключиць). Це свідчить про те, що алгоритм просканував легеневі поля, не виявив там патологічних маркерів і класифікував зображення як норму на основі чіткості анатомічних меж здорових органів.

Отже, застосування методу Grad-CAM довело, що розроблена система на базі ResNet50 володіє високою просторовою точністю та спирається на клінічно обґрунтовані візуальні ознаки захворювань.

Висновки до розділу 4

У четвертому розділі проведено емпіричне дослідження розроблених методів класифікації:

1. Експериментально доведено, що використання базових згорткових нейронних мереж без попереднього навчання не є ефективним для

вирішення задач медичної візуалізації. Модель продемонструвала низький показник повноти (60,69%) та нестабільність процесу оптимізації.

2. Встановлено, що застосування парадигми Transfer Learning (модель ResNet50) дозволяє досягти стабільної збіжності градієнтного спуску та суттєво підвищити якість класифікації. Розроблена система досягла загальної точності 91,52% та макро-повноти 91,21%, що задовольняє вимоги до медичних скринінгових алгоритмів.

3. За допомогою алгоритму Grad-CAM підтверджено клінічну релевантність розробленої системи. Візуалізація карт активацій довела, що модель здатна точно локалізувати специфічні патерни (периферичні вогнища COVID-19, масивні консолідації та дифузні зміни), нівелюючи проблему "чорного ящика".

ВИСНОВКИ

У кваліфікаційній роботі бакалавра вирішено актуальну науково-практичну задачу – досліджено та розроблено ефективні методи автоматичної мультикласової класифікації рентгенівських зображень органів грудної клітки для виявлення патологій легень. Основні результати та висновки роботи полягають у наступному:

1. Проведено системний аналіз емпіричної бази дослідження. Під час розвідувального аналізу спеціалізованого масиву COVID-19 Radiography Database (загальним обсягом 21165 зображень) доведено наявність суттєвого міжкласового дисбалансу з коефіцієнтом асиметрії IR = 7,58 на користь здорових пацієнтів. Це дозволило алгоритмічно скоригувати подальші етапи проектування системи, зокрема адаптувати цільову функцію втрат (Categorical Cross-Entropy) шляхом інтеграції пеналізаційних вагових коефіцієнтів класів.

2. Математично обґрунтовано вибір метрик оптимізації. Доведено, що в умовах структурного дисбалансу медичних даних оптимізація за загальною точністю є некоректною. Цільовим критерієм ефективності розроблених методів визначено макро-усереднену повноту, максимізація якої дозволяє мінімізувати критичні для медицини хибнонегативні прогнози (пропуск хворої людини).

3. Розроблено та досліджено базову архітектуру згорткової нейронної мережі. Експериментальне тестування мережі, що ініціалізувалася випадковими вагами, продемонструвало її нестабільність. Модель досягла максимальної точності 77,51% на 13-й епосі, однак макро-усереднена повнота склала лише 60,69%, після чого відбулася різка деградація метрик (перенавчання). Це емпірично довело, що прості CNN-архітектури не здатні формувати робастні просторові абстракції складних медичних патернів в умовах обмежених даних.

4. Розроблено та імплементовано систему на базі парадигми Transfer Learning. Для подолання проблеми згасання градієнта та високої

алгоритмічної дисперсії застосовано архітектуру ResNet50 із попередньо навченими вагами ImageNet. Завдяки "заморожуванню" згорткових блоків та навчанню власного класифікатора, вдалося досягти стабільної збіжності цільової функції. Модель продемонструвала високі експлуатаційні характеристики: загальна точність (Accuracy) склала 91,52%, точність виявлення (Precision) – 91,78%, а повнота (Recall) – 91,21%. Отримані результати підтверджують високу предиктивну ефективність розробленого методу.

5. Вирішено проблему інтерпретованості рішень ("чорного ящика"). Інтеграція алгоритму просторової локалізації Grad-CAM дозволила верифікувати результати класифікації. Аналіз згенерованих теплових карт підтвердив, що розроблена нейронна мережа приймає рішення на основі релевантних медичних маркерів: система точно локалізує периферичні зони "матового скла" при COVID-19, дифузні інтерстиціальні зміни при вірусній пневмонії та фокальні консолідації при неспецифічних помутніннях, ігноруючи при цьому здорові тканини та графічні артефакти рентгенограм.

Таким чином, усі завдання, поставлені у кваліфікаційній роботі, виконано в повному обсязі. Створений алгоритмічний конвеєр відповідає сучасним вимогам комп'ютерної радіономіки та має високий потенціал для практичного застосування у медичних інформаційних системах як засіб підтримки прийняття клінічних рішень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Wang L., Lin Z. Q., Wong A. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images // Scientific Reports. 2020. Vol. 10, no. 1. Art. 19549. DOI: https://doi.org/10.1038/s41598-020-76550-z
2. Ozturk T., Talo M., Yildirim E. A., Baloglu U. B., Yildirim O., Acharya U. R. Automated detection of COVID-19 cases using deep neural networks with X-ray images // Computers in Biology and Medicine. 2020. Vol. 121. Art. 103792. DOI: https://doi.org/10.1016/j.compbiomed.2020.103792
3. Apostolopoulos I. D., Mpesiana T. A. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks // Physical and Engineering Sciences in Medicine. 2020. Vol. 43, no. 2. P. 635–640. DOI: https://doi.org/10.1007/s13246-020-00865-4
4. Selvaraju R. R., Cogswell M., Das A. et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization // International Journal of Computer Vision. 2020. Vol. 128. P. 336–359. DOI: https://doi.org/10.1007/s11263-019-01228-7
5. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition // In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778. DOI: https://doi.org/10.48550/arXiv.1512.03385
6. Buda M., Maki A., Mazurowski M. A. A systematic study of the class imbalance problem in convolutional neural networks // Neural Networks. 2018. Vol. 106. P. 249–259. DOI: https://doi.org/10.1016/j.neunet.2018.07.011
7. Chowdhury M. E., Rahman T., Khandakar A., et al. Can AI help in screening viral and COVID-19 pneumonia? // IEEE Access. 2020. Vol. 8. P. 132665–132676. DOI: https://doi.org/10.1109/ACCESS.2020.3010287
8. Rahman T., Khandakar A., Qiblawey Y., et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images // Computers in Biology and Medicine. 2021. Vol. 132. Art. 104319. DOI: https://doi.org/10.1016/j.compbiomed.2021.104319

ДОДАТКИ

Додаток А Лістинг програмного модуля попередньої обробки та розвідувального аналізу даних

```
import os
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
def explore_dataset(base_path: str):
    """
    Виконує сканування директорій датасету, підраховує кількість зображень,
    генерує графік розподілу класів та сітку зразків рентгенограм.
    """
    # Очікувані назви класів згідно з документацією датасету
    classes = ['COVID', 'Normal', 'Lung_Opacity', 'Viral Pneumonia']
    class_counts = {}
    sample_images = {}
    print("Початок сканування датасету...\n")
    for cls in classes:
        # Враховуємо специфіку структури датасету (іноді файли лежать у підпапці 'images')
        cls_path = os.path.join(base_path, cls)
        if os.path.exists(os.path.join(cls_path, 'images')):
            cls_path = os.path.join(cls_path, 'images')
        if not os.path.exists(cls_path):
            print(f"Помилка: Директорію {cls_path} не знайдено.")
            continue
        # Пошук файлів зображень
        images = [f for f in os.listdir(cls_path) if f.lower().endswith('.png')]
        count = len(images)
        class_counts[cls] = count
        # Збереження шляху до першого зображення для візуалізації
        if count > 0:
            sample_images[cls] = os.path.join(cls_path, images[0])
        print(f"Клас '{cls}': {count} зображень")
    total_images = sum(class_counts.values())
    print(f"\nЗагальна кількість зображень: {total_images}")
    # ---------------------------------------------------------
    # 1. Побудова та збереження гістограми розподілу класів
    # ---------------------------------------------------------
    plt.figure(figsize=(10, 6))
    bars = plt.bar(class_counts.keys(), class_counts.values(),
color=['#ff9999','#66b3ff','#99ff99','#ffcc99'])
    # Додавання числових значень над стовпчиками
    for bar in bars:
        yval = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2, yval + 50, int(yval), ha='center', va='bottom',
fontsize=11)
    plt.title('Кількісний розподіл рентгенівських зображень за класами', fontsize=14, pad=15)
    plt.xlabel('Класи діагнозів', fontsize=12)
    plt.ylabel('Кількість зображень', fontsize=12)
```

```
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.savefig('class_distribution.png', dpi=300)
    print("Гістограму збережено як 'class_distribution.png'")
    # ---------------------------------------------------------
    # 2. Побудова та збереження сітки зразків зображень
    # ---------------------------------------------------------
    fig, axes = plt.subplots(1, len(sample_images), figsize=(16, 4))
    for ax, (cls, img_path) in zip(axes, sample_images.items()):
        img = mpimg.imread(img_path)
        ax.imshow(img, cmap='gray')
        ax.set_title(f"Клас: {cls}\nРозмір: {img.shape}", fontsize=12)
        ax.axis('off')
    plt.tight_layout()
    plt.savefig('sample_xrays.png', dpi=300)
    print("Зразки зображень збережено як 'sample_xrays.png'")
if __name__ == "__main__":
    # Вкажіть точний шлях до кореневої папки з класами на вашому ПК
    DATASET_PATH = r"C:\AI_projects\COVID-19_Radiography\COVID-
19_Radiography_Dataset"
    if not os.path.exists(DATASET_PATH):
        print(f"Помилка: Базовий шлях {DATASET_PATH} не існує. Перевірте правильність
шляху.")
    else:
        explore_dataset(DATASET_PATH)
```

Додаток Б Лістинг програмного модуля оптимізації базової згорткової нейронної мережі

```
import os
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,
BatchNormalization
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
from sklearn.utils.class_weight import compute_class_weight
from sklearn.model_selection import train_test_split
# Фіксуємо seed для відтворюваності (науковий стандарт)
tf.random.set_seed(42)
np.random.seed(42)
DATASET_PATH = r"C:\AI_projects\COVID-19_Radiography\COVID-
19_Radiography_Dataset"
IMG_SIZE = (224, 224)
BATCH_SIZE = 32
EPOCHS = 15 # Для базової моделі достатньо 15 епох з ранньою зупинкою
def create_dataframe(base_path):
    """Створює DataFrame виключно з оригінальних рентгенограм (без масок)"""
    classes = ['COVID', 'Normal', 'Lung_Opacity', 'Viral Pneumonia']
```

```
    filepaths = []
    labels = []
    for cls in classes:
        img_dir = os.path.join(base_path, cls, 'images')
        if os.path.exists(img_dir):
            for file in os.listdir(img_dir):
                if file.lower().endswith('.png'):
                    filepaths.append(os.path.join(img_dir, file))
                    labels.append(cls)
    df = pd.DataFrame({'filepath': filepaths, 'label': labels})
    print(f"Знайдено {len(df)} оригінальних зображень.")
    return df
def build_baseline_model(input_shape=(224, 224, 3), num_classes=4):
    """Архітектура базової згорткової нейронної мережі"""
    model = Sequential([
        # Блок 1
        Conv2D(32, (3, 3), activation='relu', input_shape=input_shape),
        BatchNormalization(),
        MaxPooling2D((2, 2)),
        # Блок 2
        Conv2D(64, (3, 3), activation='relu'),
        BatchNormalization(),
        MaxPooling2D((2, 2)),
        # Блок 3
        Conv2D(128, (3, 3), activation='relu'),
        BatchNormalization(),
        MaxPooling2D((2, 2)),
        # Класифікатор
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.5), # Регуляризація для уникнення перенавчання
        Dense(num_classes, activation='softmax') # Softmax для мультикласової класифікації
    ])
    model.compile(
        optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
        loss='categorical_crossentropy',
        metrics=['accuracy', tf.keras.metrics.Precision(name='precision'),
tf.keras.metrics.Recall(name='recall')]
    )
    return model
if __name__ == "__main__":
    # 1. Підготовка даних
    df = create_dataframe(DATASET_PATH)
    # Розбиття на Train (80%) та Validation (20%)
    train_df, val_df = train_test_split(df, test_size=0.2, stratify=df['label'], random_state=42)
    print(f"Навчальна вибірка: {len(train_df)} | Валідаційна вибірка: {len(val_df)}")
    # 2. Ініціалізація генераторів
    train_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=15,
        width_shift_range=0.1,
        height_shift_range=0.1,
```

```
    zoom_range=0.1
  )
  val_datagen = ImageDataGenerator(rescale=1./255)
  train_gen = train_datagen.flow_from_dataframe(
    dataframe=train_df, x_col='filepath', y_col='label',
    target_size=IMG_SIZE, batch_size=BATCH_SIZE, class_mode='categorical', shuffle=True
  )
  val_gen = val_datagen.flow_from_dataframe(
    dataframe=val_df, x_col='filepath', y_col='label',
    target_size=IMG_SIZE, batch_size=BATCH_SIZE, class_mode='categorical',
shuffle=False
  )
  # 3. Розрахунок ваг класів
  class_weights_array = compute_class_weight(
    class_weight='balanced',
    classes=np.unique(train_gen.classes),
    y=train_gen.classes
  )
  class_weights_dict = dict(enumerate(class_weights_array))
  # 4. Створення моделі
  model = build_baseline_model()
  model.summary()
  # 5. Налаштування колбеків (Callbacks)
  callbacks = [
    EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True, verbose=1),
    ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=2, verbose=1),
    ModelCheckpoint('baseline_cnn.keras', monitor='val_loss', save_best_only=True,
verbose=1)
  ]
  # 6. Запуск навчання
  print("\nПочаток тренування базової моделі...")
  history = model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=EPOCHS,
    class_weight=class_weights_dict,
    callbacks=callbacks
  )
  # Збереження історії навчання для графіків
  hist_df = pd.DataFrame(history.history)
  hist_df.to_csv('baseline_history.csv', index=False)
  print("\nТренування завершено! Історію збережено у 'baseline_history.csv'.")
```

Додаток В Лістинг програмного модуля класифікації на основі Transfer Learning (ResNet50)

```
import os
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout,
BatchNormalization
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
from sklearn.utils.class_weight import compute_class_weight
from sklearn.model_selection import train_test_split
# Фіксуємо seed для відтворюваності
tf.random.set_seed(42)
np.random.seed(42)
DATASET_PATH = r"C:\AI_projects\COVID-19_Radiography\COVID-
19_Radiography_Dataset"
IMG_SIZE = (224, 224)
BATCH_SIZE = 32
EPOCHS = 15
def create_dataframe(base_path):
    """Створює DataFrame виключно з оригінальних рентгенограм (без масок)"""
    classes = ['COVID', 'Normal', 'Lung_Opacity', 'Viral Pneumonia']
    filepaths = []
    labels = []
    for cls in classes:
        img_dir = os.path.join(base_path, cls, 'images')
        if os.path.exists(img_dir):
            for file in os.listdir(img_dir):
                if file.lower().endswith('.png'):
                    filepaths.append(os.path.join(img_dir, file))
                    labels.append(cls)

    df = pd.DataFrame({'filepath': filepaths, 'label': labels})
    return df
def build_resnet_model(input_shape=(224, 224, 3), num_classes=4):
    """Побудова архітектури на базі попередньо навченої ResNet50"""
    # Завантаження базової моделі без верхнього класифікатора (include_top=False)
    base_model = ResNet50(weights='imagenet', include_top=False, input_shape=input_shape)
    # Заморожуємо ваги базової моделі (Feature Extraction phase)
    base_model.trainable = False
    model = Sequential([
        base_model,
        GlobalAveragePooling2D(), # Зменшення розмірності замість Flatten для уникнення
перенавчання
        BatchNormalization(),
        Dense(256, activation='relu'),
        Dropout(0.5), # Регуляризація
        Dense(num_classes, activation='softmax')
    ])
    model.compile(
        optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
        loss='categorical_crossentropy',
        metrics=['accuracy', tf.keras.metrics.Precision(name='precision'),
tf.keras.metrics.Recall(name='recall')]
    )
```

```
    return model
if __name__ == "__main__":
    print("Підготовка даних для ResNet50...")
    df = create_dataframe(DATASET_PATH)
    train_df, val_df = train_test_split(df, test_size=0.2, stratify=df['label'], random_state=42)
    # ЗВЕРНІТЬ УВАГУ: замість rescale=1./255 використовується спеціальний
preprocess_input
    train_datagen = ImageDataGenerator(
        preprocessing_function=preprocess_input,
        rotation_range=15,
        width_shift_range=0.1,
        height_shift_range=0.1,
        zoom_range=0.1
    )
    val_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
    train_gen = train_datagen.flow_from_dataframe(
        dataframe=train_df, x_col='filepath', y_col='label',
        target_size=IMG_SIZE, batch_size=BATCH_SIZE, class_mode='categorical', shuffle=True
    )
    val_gen = val_datagen.flow_from_dataframe(
        dataframe=val_df, x_col='filepath', y_col='label',
        target_size=IMG_SIZE, batch_size=BATCH_SIZE, class_mode='categorical',
shuffle=False
    )
    # Розрахунок ваг класів для збалансування функції втрат
    class_weights_array = compute_class_weight(
        class_weight='balanced',
        classes=np.unique(train_gen.classes),
        y=train_gen.classes
    )
    class_weights_dict = dict(enumerate(class_weights_array))
    # Створення моделі
    model = build_resnet_model()
    print("\nАрхітектура Transfer Learning моделі:")
    model.summary()
    # Callbacks
    callbacks = [
        EarlyStopping(monitor='val_loss', patience=4, restore_best_weights=True, verbose=1),
        ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=2, verbose=1),
        ModelCheckpoint('resnet50_model.keras', monitor='val_loss', save_best_only=True,
verbose=1)
    ]
    print("\nПочаток тренування ResNet50 (Feature Extraction)...")
    history = model.fit(
        train_gen,
        validation_data=val_gen,
        epochs=EPOCHS,
        class_weight=class_weights_dict,
        callbacks=callbacks
    )
    # Збереження історії
    hist_df = pd.DataFrame(history.history)
```

```
    hist_df.to_csv('resnet_history.csv', index=False)
    print("\nТренування завершено! Історію збережено у 'resnet_history.csv'.")
```

Додаток Г Лістинг програмного модуля інтерпретації результатів (Grad-CAM)

```
import os
import cv2
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras.applications.resnet50 import preprocess_input
# Вимикаємо зайві логи TF
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
DATASET_PATH = r"C:\AI_projects\COVID-19_Radiography\COVID-
19_Radiography_Dataset"
MODEL_PATH = "resnet50_model.keras"
IMG_SIZE = (224, 224)
CLASSES = ['COVID', 'Lung_Opacity', 'Normal', 'Viral Pneumonia']
def get_img_array(img_path, size):
    img = tf.keras.preprocessing.image.load_img(img_path, target_size=size)
    array = tf.keras.preprocessing.image.img_to_array(img)
    array = np.expand_dims(array, axis=0)
    return preprocess_input(array)
def make_gradcam_heatmap(img_array, model, last_conv_layer_name):
    """Генерація теплової карти Grad-CAM для вкладеної моделі ResNet50"""
    # Витягуємо базову модель (ResNet50) та її останній згортковий шар
    base_model = model.layers[0]
    last_conv_layer = base_model.get_layer(last_conv_layer_name)
    # 1. Модель від входу до останнього згорткового шару
    last_conv_layer_model = tf.keras.Model(base_model.inputs, last_conv_layer.output)
    # 2. Модель класифікатора (всі шари ПІСЛЯ базової моделі)
    classifier_input = tf.keras.Input(shape=last_conv_layer.output.shape[1:])
    x = classifier_input
    for layer in model.layers[1:]:
        x = layer(x)
    classifier_model = tf.keras.Model(classifier_input, x)
    # Обчислення градієнтів
    with tf.GradientTape() as tape:
        last_conv_layer_output = last_conv_layer_model(img_array)
        tape.watch(last_conv_layer_output)
        preds = classifier_model(last_conv_layer_output)
        top_pred_index = tf.argmax(preds[0])
        top_class_channel = preds[:, top_pred_index]
    # Градієнт цільового класу відносно виходу згорткового шару
    grads = tape.gradient(top_class_channel, last_conv_layer_output)
    pooled_grads = tf.reduce_mean(grads, axis=(0, 1, 2))
    last_conv_layer_output = last_conv_layer_output[0]
    heatmap = last_conv_layer_output @ pooled_grads[..., tf.newaxis]
    heatmap = tf.squeeze(heatmap)
    heatmap = tf.maximum(heatmap, 0) / tf.math.reduce_max(heatmap)
```

```
    return heatmap.numpy()
def save_and_display_gradcam(img_path, heatmap, output_path, title="Grad-CAM"):
    img = cv2.imread(img_path)
    img = cv2.resize(img, IMG_SIZE)
    # Масштабування теплової карти
    heatmap = np.uint8(255 * heatmap)
    jet = cv2.applyColorMap(heatmap, cv2.COLORMAP_JET)
    # Накладання з альфа-каналом (прозорістю)
    superimposed_img = cv2.addWeighted(img, 0.6, jet, 0.4, 0)
    # Створення графіку з двох частин (Оригінал + Grad-CAM)
    fig, ax = plt.subplots(1, 2, figsize=(10, 5))
    ax[0].imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    ax[0].set_title(f"Оригінал: {title}")
    ax[0].axis('off')
    ax[1].imshow(cv2.cvtColor(superimposed_img, cv2.COLOR_BGR2RGB))
    ax[1].set_title(f"Grad-CAM: {title}")
    ax[1].axis('off')
    plt.tight_layout()
    plt.savefig(output_path, dpi=300)
    plt.close()
if __name__ == "__main__":
    print("Завантаження навченої моделі...")
    model = tf.keras.models.load_model(MODEL_PATH)
    # Остання згортка в архітектурі ResNet50
    conv_layer_name = "conv5_block3_out"
    print("Генерація Grad-CAM для зразків кожного класу...")
    for cls in CLASSES:
        img_dir = os.path.join(DATASET_PATH, cls, 'images')
        if not os.path.exists(img_dir): continue
        # Беремо перше зображення з класу
        sample_img = os.listdir(img_dir)[0]
        img_path = os.path.join(img_dir, sample_img)
        img_array = get_img_array(img_path, IMG_SIZE)
        try:
            heatmap = make_gradcam_heatmap(img_array, model, conv_layer_name)
            out_file = f"gradcam_{cls}.png"
            save_and_display_gradcam(img_path, heatmap, out_file, title=cls)
            print(f"Збережено візуалізацію: {out_file}")
        except Exception as e:
            print(f"Помилка при генерації для {cls}: {e}")
    print("\nАналіз Grad-CAM успішно завершено! Перевірте збережені файли
(gradcam_*.png).")
```