

Вінницький національний технічний університет  
(повне найменування вищого навчального закладу)  
Факультет інтелектуальних інформаційних технологій та автоматизації  
(повне найменування інституту, назва факультету (відділення))  
Кафедра автоматизації та інтелектуальних інформаційних технологій  
(повна назва кафедри (предметної, циклової комісії))

## **Магістерська кваліфікаційна робота на тему:**

**«РОЗРОБКА ГЛИБОКОЇ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ДЛЯ  
КЛАСИФІКАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ»**

Виконав: студент 5 курсу, групи 1АКІТ-21м  
спеціальності 151 – Автоматизація та  
комп'ютерно-інтегровані технології  
(шифр і назва напряму підготовки, спеціальності)

Сушко Д.В.

(прізвище та ініціали)

Керівник: к.т.н., доцент, доцент каф. АІТ

Кулик Я.А.

(прізвище та ініціали)

«    » \_\_\_\_\_ 2022 р.

Консультант: к.т.н., доцент, зав. каф. КСУ

Ковтун В.В.

Рецензент:

\_\_\_\_\_  
(прізвище та ініціали)

«    » \_\_\_\_\_ 2022 р.

Допущено до захисту  
в.о. зав. кафедри АІТ

Бісікало О.В.

«    » \_\_\_\_\_ 2022 р.

**Вінницький національний технічний університет**  
 Факультет інтелектуальних інформаційних технологій та автоматизації  
 Кафедра автоматизації та інтелектуальних інформаційних технологій  
 Рівень вищої освіти другий (магістерський)  
 Галузь знань 15 – Автоматизація та приладобудування  
 (шифр і назва)  
 Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології  
 (шифр і назва спеціальності)  
 Освітньо-професійна програма Інтелектуальні комп'ютерні системи  
 (назва освітньо-професійної програми)

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри АІТ**  
 д.т.н., проф. Бісікало О. В.

\_\_\_\_\_ 2022 року

## ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Сушку Денису Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка глибокої нейромережевої системи для класифікації медичних зображень»,

Керівник роботи Кулика Я.А., доцент, доцент кафедри АІТ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “\_\_” \_\_\_\_\_ 202\_ року  
 № \_\_\_\_\_

2. Термін подання студентом роботи 16.12.2022 р.

3. Вихідні дані до роботи:

Рентгенографічне зображення легенів роздільної здатності не менше 512 x 512; кількість прихованих шарів – не більше 1000; швидкість розпізнавання не повинна 1 хв; номінальна напруга живлення +5 В ±5%; точність розпізнавання – не менше 80%.

4. Зміст текстової частини

Вступ. Аналіз предметної області. Моделі розпізнавання вірусних захворювань легенів з використанням згорткових нейромереж. Обчислювальний експеримент на основі згорткових нейромереж для задач розпізнавання медичних зображень. Висновки. Перелік посилань. Додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема даних. Зразок рентгенографії знімків. Порівняння методів  
Характеристики моделі.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Виконання прийняв
Спеціальна частина	к.т.н., доцент, доцент кафедри АПТ Кулик Я.А.		
Економічний розділ	Козловський В.О, к.е.н., професор кафедри ЕПОВ		

7. Дата видачі завдання 16.09.2022 р.**КАЛЕНДАРНИЙ ПЛАН**

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1.	Вибір, узгодження та затвердження теми МКР	03.09.2021	06.03.2022	
2.	Аналіз предметної області	07.09.2022	15.09.2022	
3.	Моделі розпізнавання вірусних захворювань легенів з використанням згорткових нейромереж	16.09.2022	29.09.2022	
4.	Обчислювальний експеримент на основі згорткових нейромереж для задач розпізнавання медичних зображень	30.09.2022	19.10.2022	
7.	Моделювання нейромережі	11.10.2022	17.10.2022	
9.	Оформлення пояснювальної записки та графічної частини	25.11.2022	01.12.2022	
10.	Нормоконтроль	02.12.2022	09.12.2022	
11.	Попередній захист БДР, доопрацювання, рецензування БДР	12.12.2022	16.12.2022	
12.	Захист БДР ЕК	___.12.2022	___.12.2022	

Студент

(підпис)

Сушко Д.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Кулик Я.А.

(прізвище та ініціали)

## АНОТАЦІЯ

УДК 004.514, 004.522

3 рентгенівських знімки, 5 рисунків, 8 таблиць, 8 графіків

Зараз світ перебуває у розпалі пандемії, викликаній поширенням різних штамів коронавірусної інфекції. Вірус COVID та SARS утворює багато слизу в дихальній системі та спричиняє набряк дихальних шляхів. Це може призвести до смерті. Зараз у боротьбі з новим коронавірусом сформовано два напрямки: виявлення та лікування.

У першому розділі кваліфікаційної роботи описано процес розвитку епідеміологічної загрози, відображено стан досліджень у галузі обробки медичних зображень, а також проведено наукові дослідження щодо використання CNN для обробки медичних зображень в умовах пандемії.

У другому розділі кваліфікаційної роботи описано раннє виявлення COVID та SARS з використанням CNN, Розглянуто згорткові нейронні мережі, освітлено капсульні мережі, досліджено гібридну модель CNN та нейромережну архітектуру VGG-CapNet.

У третьому розділі кваліфікаційної роботи описано обчислювальний експеримент, представлено і потім проаналізовано результати моделювання, досліджено модель CNN-CapsNet і наведено результати використання VGG-CapsNet для виявлення COVID і SARS.

Ключові слова: COVID, SARS, нейронні мережі, дослідження, обробка, моделювання.

## ANNOTATION

UDC 004.514, 004.522

3 x-rays, 5 drawings, 8 tables, 8 graphs

The world is currently in the midst of a pandemic caused by the spread of various strains of the coronavirus infection. The COVID and SARS virus produces a lot of mucus in the respiratory system and causes the airways to swell. This can lead to death. Currently, two directions have been formed in the fight against the new coronavirus: detection and treatment.

The first chapter of the qualification work describes the development process of the epidemiological threat, reflects the state of research in the field of medical image processing, and also conducts scientific research on the use of CNNs for processing medical images in pandemic conditions.

In the second part of the qualification work, the early detection of COVID and SARS using CNN is described, convolutional neural networks are considered, capsule networks are illuminated, the CNN hybrid model and the VGG-CapNet neural network architecture are investigated.

The third section of the qualification paper describes the computational experiment, presents and then analyzes the simulation results, examines the CNN-CapsNet model, and gives the results of using VGG-CapsNet to detect COVID and SARS.

Keywords: COVID, SARS, neural networks, research, processing, modeling.

## ЗМІСТ

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Актуальні бактерологічні загрози перед людством .....	11
1.2 Стан досліджень в сфері розпізнавання медичних зображень .....	12
1.3 CNN як засіб розпізнавання медичних зображень .....	13
2 МОДЕЛІ РОЗПІЗНАВАННЯ ВІРУСНИХ ЗАХВОРЮВАНЬ ЛЕГЕНІВ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ .....	23
2.1 Раннє виявлення COVID та SARS з використанням CNN.....	23
2.2 Згорткові нейромережі .....	26
2.3 Гібридна модель архітектури CNN.....	27
2.4 Капсульні мережі .....	32
2.5 Нейромережева модель VGG-CapNet .....	35
2.6 Вибір датасету.....	38
3 ОБЧИСЛЮВАЛЬНИЙ ЕКСПЕРИМЕНТ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ ДЛЯ ЗАДАЧ РОЗПІЗНАВАННЯ МЕДИЧНИХ ЗОБРАЖЕНЬ.....	40
3.1 Обчислювальний експеримент і результати моделювання .....	40
3.2 Модель CNN-CapsNet для виявлення COVID та SARS .....	41
3.3 Модель VGG-CapsNet для виявлення COVID та SARS.....	47
3.4 Аналіз результатів моделювання .....	53
4. ЕКОНОМІЧНИЙ РОЗДІЛ .....	58
4.1 Технологічний аудит розробленої нейромережевої системи для класифікації медичних зображень .....	58
4.2 Розрахунок витрат на розробку нейромережевої системи для класифікації медичних зображень .....	62
4.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки..	66
ВИСНОВКИ .....	73
СПИСОК ДЖЕРЕЛ .....	74
ДОДАТОК А .....	82
ДОДАТОК Б.....	85
ДОДАТОК В.....	107

## В І Д Г У К

на магістерську кваліфікаційну роботу “ Розробка глибокої нейромережевої системи для класифікації медичних зображень ”

студента ФІТА групи 1АКІТ-21м

Сушка Дениса Вікторовича

Раннє виявлення коронавірусу допоможе зупинити поширення пандемії. На підставі аналізу рентгенограм грудної клітки пацієнтів, інфікованих COVID та SARS медичні зображення були оброблені для виявлення випадків зараження. В умовах епідемії автоматичне виявлення хвороби є актуальним напрямком сучасних досліджень.

В роботі проводиться аналіз існуючих моделей розпізнавання хвороби covid-19 на основі рентгенографічних знімків легень пацієнтів. Навчена модель порівнювалась з існуючими за різними критеріями точності. Були проведені експерименти, які показали хорошу точність навченої моделі на перевірочні вибірці.

При виконанні магістерської дипломної роботи студент показав достатній рівень інженерної підготовки та хорошу ерудицію. Він орієнтується в питаннях штучного інтелекту, статистики, оцінки точності, підбору даних, розроблення програмного забезпечення, вивчення нових програмних інструментів.

Переважну більшість поставлених задач Сушко Д. В. вирішував самостійно; показав свою здатність до засвоєння нових знань, зокрема в області машинного навчання та оцінки його якості.

Проявив дисциплінованість та обов'язковість в роботі, виявив уміння здебільшого вчасно виконувати поставлені задачі, завдяки чому графік виконання магістерської дипломної роботи ним витримувався з мінімальними порушеннями.

Проте роботі притаманні **деякі недоліки**. Зокрема, не оцінено обчислювальну складність нейромережі, не розглянуто питання швидкодії. Також не до кінця пророблені питання оцінювання інших характеристик нейромережі-повнота, відгук, а також база не оцінювалась на різноманітність даних з метою запобігання перенавчання. Також у здобувача були деякі проблеми з розумінням стандартів оформлення роботи.

В цілому магістерську дипломну роботу виконано на достатньо високому рівні і вона заслуговує оцінку “А”, а Сушко Денис Вікторович – присвоєння ступеня вищої освіти магістр зі спеціальності 151- «Автоматизація та комп'ютерно-інтегровані технології» , освітня програма «Інтелектуальні комп'ютерні системи».

Керівник магістерської дипломної роботи  
к.т.н., доцент кафедри АІТ

Кулик Я.А.

## Відгук опонента на магістерську кваліфікаційну роботу

студента Сушко Дениса Вікторовича група 1АКІТ-21м  
на тему: Розробка глибокої нейромережевої системи для класифікації медичних зображень.

Актуальність роботи в контексті спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» доведена результатами інформаційного пошуку та аналізу літературних джерел. Додатковим підтвердженням актуальності роботи слугують опубліковані тези на Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації.

Перший розділ магістерської кваліфікаційної роботи повністю присвячений огляду літературних та інформаційних джерел за обраною темою. Проаналізовано не менш ніж три аналоги створеної системи. Функціональність та форм-фактор створеної системи цілком визначені на основі результатів критичного огляду літератури.

Прийняті рішення обґрунтовані результатами огляду літератури, результатами проектування та втілені в функціонуючу програмну систему. Результати її тестування доводять правильність прийнятих рішень.

Експериментальні дослідження продумані і повні. Тести охоплюють як функції інтерфейсу створеної системи, так і доводять якість та повноту виконання нею функціонального призначення, обґрунтованого на етапі проектування.

Вміст графічної частина повною мірою репрезентує всі отримані в магістерській кваліфікаційній роботі результати. Якість рисунків в графічній частині прийнятна.

**Недоліки:** Рішення щодо вибору нейромереж для здійснення класифікації бажано було обґрунтувати детальніше. Не всі коефіцієнти наведених формул зрозуміло описані. Питанню вибору даних тесту для тестування створеної системи варто біло відвести окремий підрозділ.

Загалом магістерська кваліфікаційна робота відповідає спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», заслуговує на оцінку \_\_\_\_\_, а її автор заслуговує присудження кваліфікації: ступінь вищої освіти магістр, спеціальність «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні комп'ютерні системи».

**Опонент на магістерську кваліфікаційну роботу**

Доцент кафедри КН

\_\_\_\_\_

Колесницький О.К.



## ВСТУП

Актуальність теми. Зараз світ перебуває у розпалі пандемії, викликаній поширенням різних штамів коронавірусної інфекції. Вірус COVID та SARS утворює багато слизу в дихальній системі та спричиняє набряк дихальних шляхів. Якщо слиз не видалити, це може призвести до величезних відбитків у легенях. Це може призвести до смерті [1]. Станом на осінь 2021 року від COVID та SARS померло понад 5 мільйонів людей [2]. Зараз у боротьбі з новим коронавірусом сформовано два напрямки: виявлення та лікування. Раннє виявлення коронавірусу допоможе зупинити поширення пандемії. На підставі аналізу рентгенограм грудної клітки пацієнтів, інфікованих COVID та SARS, встановлено, що на ранніх стадіях захворювання спостерігаються слабкі симптоми, які супроводжуються легкими помутніннями в легенях та утворенням лімфовузлів. , медичні зображення були оброблені для виявлення великих випадків зараження в умовах епідемії є актуальним напрямком сучасних досліджень.

Мета і задачі дослідження. Метою даної акредитації для освітнього рівня «Магістр» є підвищення рівня цілісності подання медичної графічної інформації в умовах пандемії. Для досягнення поставлених цілей необхідно виконати наступні завдання:

- Проаналізувати стан досліджень в даній предметній області.
- Дослідити існуючі на даний час практичні реалізації на основі згорткових нейронних мереж.
- Проаналізувати моделі виявлення інфекційних захворювань з використанням згорткових нейронних мереж.
- Дослідити обчислювальні експерименти на основі згорткових нейронних мереж для задач опрацювання медичних зображень в умовах пандемії.

Об'єкт дослідження процесу опрацювання медичних зображень в умовах пандемії.

Предмет дослідження методи аналітичного опрацювання медичних

зображень в умовах пандемії.

Наукова новизна отриманих результатів ідентифікаційної роботи полягає в подальшому розвитку моделі згорткової капсульної нейронної мережі та моделі візуальної геометричної капсульної мережі.

Практичне значення отриманих результатів. Виконується проектування та створення прототипів моделей нейронної мережі Convolution Capsule і Visual Geometric Capsule Network.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Актуальні бактерологічні загрози перед людством

В даний час, за даними «Національного центру біотехнологічної інформації» (NCBI), існує 219 вірусів, які можуть інфікувати людину [3]. Коронавіруси — це група вірусів, які в основному викликають легкі проблеми з кашлем і симптомами застуди. Більшість коронавірусів нешкідливі для людини. Однак новий коронавірус COVID та SARS дуже агресивний. Часто це призводить до смерті пацієнтів.

Вірус називають коронавірусом, тому що на його поверхні є білкова структура, схожа на коронавірус. Коронавіруси — одноланцюгові РНК-віруси. Їх мутації сильніші за віруси на основі ДНК. COVID та SARS поширюється швидше, ніж інші вірусні ГРВІ. При кожному контакті COVID та SARS з тілом людини відбувається дуже сильний контакт з клітинною мембраною людини через білкові спайки. Інкубаційний період у коронавірусів триває приблизно 5 днів [4].

Можливість виявлення коронавірусу на ранній стадії допоможе зупинити поширення пандемії. У країнах, що розвиваються, медичні послуги є недоступними та дорогими. Тому існують фінансові обмеження, пов'язані з вартістю процедур діагностики та тестування. В даний час використовується великий перелік аналітичних методів [5] та інструментів [6], які можуть бути ефективно застосовані в пандемічних ситуаціях.

У зв'язку з поширенням COVID та SARS у різних країнах з квітня 2020 року різко зросла кількість рентгенів грудної клітки та комп'ютерної томографії [7]. Розвиток захворювання супроводжується посиленням симптомів. Це викликає великий туман навколо вузла. Додатково в обох легенях утворюються множинні ущільнення. Навколо вузлика утворюється ореол або запаморочення та кластерні ознаки. У важкій стадії множинні дифузні ураження легень. Це в свою чергу призводить до формування легеневого фіброзу. В результаті рентгенівське зображення легень стає білим

[8]. Це спонукало наукове співтовариство до розробки систем, які використовують обробку аналізу рентгенівських зображень для діагностики COVID та SARS.

## **1.2 Стан досліджень в сфері розпізнавання медичних зображень**

У науковій літературі описано великий перелік досліджень на основі систем штучного інтелекту для різних медичних цілей. Зокрема, діагностика хвороби Альцгеймера, оцінка стадії перебігу раку, аналіз та обробка результатів біопсії та дерматоскопії тощо [9]. За останній період пандемія COVID та SARS створила величезний тиск на роботу медичного персоналу [10]. Тому формування системи медичної діагностики та лікування на основі штучного інтелекту є практичним напрямком досліджень.

Інформаційні системи на основі методів аналізу даних, включаючи методи машинного та глибокого навчання, для обробки медичних зображень. Моделі глибокого навчання використовують методи фільтрації та класифікації, подібні до людського мозку [11]. Глибоке навчання — це підкатегорія машинного навчання, яка побудована на кількох рівнях обробки даних. Для аналізу та класифікації вхідних даних використовуються окремі шари. Дані можна вводити в текстовому, звуковому або графічному вигляді. Таким чином, глибоке навчання може бути ефективно застосоване до проблем класифікації медичних зображень [12].

Згорточні нейронні мережі (CNN) найкраще підходять для глибокої класифікації зображень. Але наразі існують певні концептуальні обмеження. Зокрема, під час роботи CNN з максимальним об'єднанням, інформація про місцезнаходження об'єкта, яку мережа використовує для ідентифікації, втрачається. Крім того, CNN не враховують численні просторові зв'язки між простими об'єктами. Щоб подолати ці концептуальні обмеження CNN, автори використовували згорточні капсульні мережі (CapsNet) для класифікації рентгенівських зображень [13].

У [14] автори запропонували модель глибокого навчання на основі Visual

Geometry Group Capsule Network (VGG-CapsNet) для отримання більш детальної інформації з рентгенівських зображень шляхом поєднання VGGNet і CapsNet.

### **1.3 CNN як засіб розпізнавання медичних зображень**

Апостолопулос [15] запропонував модель виявлення COVID та SARS шляхом аналізу рентгенівських знімків грудної клітки. Модель розроблена на основі глибокого навчання з використанням CNN, особливості передачі процесу навчання. Для навчання моделі автори використовували рентгенівські зображення грудної клітини здорових людей, інфікованих пневмонією людини, і пацієнтів, інфікованих COVID та SARS. Ці зображення є загальнодоступними в сховищах медичних даних. Автори розробили та протестували модель за допомогою двох наборів рентгенівських зображень. Кожен використаний набір даних містить рентгенівські знімки понад 220 пацієнтів з інфекцією COVID та SARS, понад 700 пацієнтів із пневмонією та понад 500 здорових громадян. Проаналізувавши результати моделювання, автори публікації виявили, що модель досягла рівня точності 96,78% при пошуку пацієнтів, інфікованих COVID та SARS. У майбутньому автори планують включити в навчальний процес колекції та колекції рентгенівських знімків пацієнтів, інфікованих ГРВІ, Ебола тощо. Хол в [16] обговорює важливість аналізу та обробки рентгенівських знімків грудної клітки для ранньої діагностики та виявлення COVID та SARS. Автори пропонують модель глибокого навчання, сформовану на основі комбінації попередньо навчених мереж на основі VGG16, Resnet50 і CNN. Під час навчання та тестування було використано відкриту колекцію рентгенограм органів грудної клітки понад 130 пацієнтів із інфекцією COVID та SARS та понад 300 пацієнтів із звичайними ГРВІ та пневмонією. Автори попередньо підготували Resnet50, використовуючи понад сотню рентгенівських зображень пацієнтів з COVID та SARS і понад сотню рентгенівських зображень пацієнтів із пневмонією. Запропонована авторами комбінація, заснована на трьох різних класифікаторах, була протестована на понад 30 невидимих

рентгенівських зображеннях пацієнтів з COVID та SARS і понад 200 зображеннях пацієнтів з ГРВІ та пневмонією. Автори стверджують, що точність моделі суміші становить 91,24%.

Робота Zhang [17] спрямована на підвищення ефективності обробки рентгенівських знімків грудної клітки для виявлення пацієнтів, інфікованих COVID та SARS. Тому що вірус позитивно впливає на органи дихання, особливо на легені. Автори пропонують модель виявлення легеневих аномалій на основі глибокого навчання. У дослідженні було використано сто рентгенівських знімків грудної клітини, доступних на Github. У досліджуваному наборі даних було отримано 70 рентгенівських зображень грудної клітки для пацієнтів з COVID та SARS. Решта 30 зображень отримано для хворих на ГРВІ, пневмонію тощо. Для ефективного використання можливостей методів, заснованих на глибокому навчанні, автори використали понад тисячу зображень пацієнтів із пневмонією. Опубліковані результати моделювання забезпечили 96% точність ідентифікації COVID та SARS, тоді як у випадках, не пов'язаних з COVID та SARS, точність становила 70,65%. Акар і Коркмаз [18] обговорили процес виявлення COVID та SARS. Автори використовують конструкцію на основі штучного інтелекту для виявлення COVID та SARS на основі рентгенівського аналізу грудної клітки. Squeezenet і Bayesian оптимізація використовуються для побудови моделей глибокого навчання. Для підвищення точності використовуються гіперпараметри та розширені набори даних. Автори заявляють про 98,3% точності виявлення COVID та SARS, пневмонії та звичайних випадків. Khalifa [19] вивчав процес виявлення запалення легенів, викликаного COVID та SARS. Автори описують використання Generative Adversarial Networks (GAN) з моделями глибокого навчання передачі «AlexNet», «Resnet18», «Squeezenet» і «GoogleNet». Для навчання моделі та динамічного тестування використовувався набір даних із понад 5800 рентгенограм грудної клітки інфікованих пневмонією або здорових пацієнтів. На підставі результатів моделювання автори прийшли до висновку, що комбінація GAN і "Resnet18" забезпечує 99% точності відповідно до точності, запам'ятовування та оціночної шкали (F-1). Салман [20] опублікував

дослідження про рентгенівські зображення високої чіткості, щоб зменшити навантаження на радіологів для виявлення COVID та SARS. Було використано набір із 260 зображень, доступних на Kaggle та Github. Розробити моделі глибокого навчання на основі CNN. Використаний набір зображень містить рентгенівські знімки грудної клітки 130 здорових громадян і 130 пацієнтів, інфікованих COVID та SARS. Автори стверджують, що в результаті моделювання були досягнуті цільові показники точності, еквівалентні радіологу.

Група авторів під керівництвом Sethi [21] запропонувала використовувати алгоритми глибокого навчання для виявлення COVID та SARS на основі аналізу рентгенограм грудної клітки. Автори пропонують гібридний підхід завдяки поєднанню SVM і «resnet50». Використовувалися рентгенівські зображення грудної клітини, доступні на Github і Kaggle. Видання заявляє про точність 95,38%. Maghdid [22] зосередився на формуванні загальнодоступного набору даних рентгенограм грудної клітки. Автори розробили метод діагностики COVID та SARS за допомогою трансферного навчання на основі моделей глибокого навчання. Розробив дві моделі прогнозування на основі "Alexnet", використовуючи CNN. В опублікованому матеріалі модель CNN на основі потоку досягає 94,1% точності, а модель CNN на основі Alexnet досягає 98% точності.

Bassi and Attucks [23] запропонували модель класифікації рентгенограм грудної клітки для виявлення COVID та SARS. В основі моделі лежить

«DenseNet121 CNN» використовує рентгенівські зображення грудної клітки пацієнтів з COVID та SARS, хворих на пневмонію та здорових громадян. Модель, описана в статті, пройшла два етапи навчання. На першому етапі для навчання використовується набір зображень. На другому етапі навчання базується на базі рентгенограм грудної клітини. Виходячи з результатів симуляційного дослідження, автори повідомляють, що тест на COVID та SARS був точним на 97,8 відсотка.

Oztark [24] розробив засновану на глибокому навчанні модель для виявлення COVID та SARS на основі обробки рентгенівських знімків грудної

клітки. В опублікованих результатах використовувався класифікатор "Darknet CNN", що складається з 17 шарів і системи виявлення об'єктів

"Йоло". Запропонована модель може бути успішно використана для бінарної та мультикласифікації рентгенограм грудної клітки. У процесі виявлення пацієнтів з інфекцією COVID та SARS точність бінарної класифікації склала 98,08%, а мультикласифікації – 87,02%.

У статті [25] обговорюється роль обробки аналізу зображень КТ грудної клітки у виявленні COVID та SARS. Проаналізовано обмеження КТ грудної клітини у виявленні COVID та SARS. У дослідженні використовувалися алгоритми на основі штучного інтелекту. Автори розробили інформаційну систему для комплексної діагностики COVID та SARS на основі клінічних симптомів, лабораторних даних і даних КТ грудної клітки. Було розроблено три різні моделі. Перша модель була сформована на основі методів глибинного навчання та реалізована за допомогою CNN. Модель тренується тільки на КТ грудної клітини. Друга модель формується опорними векторними машинами та алгоритмами випадкового лісу. Модель була навчена на доступній клінічній інформації. Третя модель формується на основі CNN. Вона навчається на основі поєднання клінічної та радіологічної інформації. Автори протестували три запропоновані моделі для збору даних понад 270 пацієнтів. Для третьої моделі була досягнута площа під кривою 0,92.

Лі [26] розробив засновану на глибокому навчанні модель «COVNet» для виявлення COVID та SARS за зображеннями КТ грудної клітини. Для навчання та тестування моделі було використано понад 4300 КТ-зображень грудної клітини понад 3300 пацієнтів. Автори створили вказані набори даних з 2016 по 2020 рік. Зображення, зібрані для пацієнтів з COVID та SARS, хворих на пневмонію та здорових людей. Автори стверджують, що площа під кривою становить 0,96 для КТ зображень COVID та SARS і 0,95 для громадян з пневмонією.

Тораман [27] досліджував проблему виявлення COVID та SARS за симптомами, схожими на легеневу інфекцію. Проаналізовано фактори, що сприяють розробці комп'ютерних засобів діагностики COVID та SARS. Автори



вказують на роль і важливість інформації, інтегрованої в рентгенівські знімки, для виявлення COVID та SARS. Використовується модель «CapsNet», заснована на глибокому навчанні. Розроблена система виявлення COVID та SARS була навчена та практично перевірена на рентгенівських знімках органів грудної клітки. Оприлюднено понад 230 рентгенівських знімків хворих на COVID та SARS, понад 1000 рентгенівських знімків здорових громадян і понад 1000 рентгенівських знімків хворих на пневмонію. Видання зазначає, що при виявленні COVID та SARS двійкова класифікація досягла 97,24% точності, а мультикласова – 84,22%.

Шойбі [28] запропонував характеристики вірусу COVID та SARS. Автори досліджують використання методів штучного інтелекту, включаючи глибоке та машинне навчання, для виявлення COVID та SARS. Примітно, що глибоке навчання є більш популярним, ніж методи на основі машинного навчання. Це завдяки автоматизації процесу вибору ознак, відбору та класифікації. У цій статті детально розглядаються моделі наукового дослідження, сформовані на основі глибокого навчання. Ці моделі використовуються для виявлення COVID та SARS під час аналітичної обробки рентгенівських зображень грудної клітки. Крім того, автори аналізують проблему розробки системи діагностики COVID та SARS на основі методів глибинного навчання.

Хан і Аслам [29] представили інструмент, заснований на підході глибокого навчання для діагностики COVID та SARS за рентгенівськими знімками. У цій роботі досліджуються деякі моделі глибокого навчання, зокрема, повідомляється, що моделі «VGG16» і «VGG-19» демонструють вищу продуктивність порівняно з іншими моделями, які зараз використовуються.

Аман Джайсвал [30] використовував аналітичну обробку рентгенівських зображень для діагностики COVID та SARS на основі алгоритмів глибокого навчання. У статті порівнюються характеристики архітектур CNN і пропонується правило більшості. Опубліковані результати досліджень дозволяють гарантувати оптимальні показники на рівні 98,96%. Нур, Комерт і Полат [31] описують систему автоматичної діагностики на основі архітектури CNN для виявлення позитивних випадків зараження COVID та SARS за

допомогою рентгенівських зображень. Запропонована модель CNN сформована на основі п'яти мереж згорткової послідовності. Модель CNN, навчена з нуля, виділяє дискримінаційні особливості алгоритму

Машинне навчання k-NN, SVM і DT. Автори відзначають найбільш ефективний результат для класифікатора SVM з точністю 98,97%. Група авторів під керівництвом Чоудхурі [32] досліджувала метод глибокого перенесення на основі формування CNN (DCNN) для автоматичного виявлення пневмонії, викликані COVID та SARS, за допомогою рентгенівських зображень. Дослідники підготували вісім різних популярних алгоритмів, використовуючи Глибоке навчання на основі CNN. Серед них «SqueezeNet», «ResNet18», «InceptionV3» тощо. Дослідження трьох категорій з використанням доповнення даних забезпечило найвищу точність класифікації 97,94%.

Aslan [33] опублікував детальну інформацію про бази даних радіологічних зображень, сегментацію легень, аналіз даних та архітектуру для класифікації. Запропонована авторами узагальнена блок-схема аналітичної обробки даних наведена на рисунку 1.1.

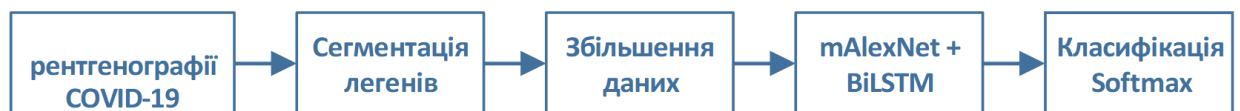


Рисунок 1.1 – Схема аналітичного опрацювання візуальних даних [33]

Зокрема, зразок рентгенографічного зображення для інфікованого COVID та SARS пацієнта подано на рисунку 1.2.



Рисунок 1.2 – Зразок рентгенографічного зображення легенів інфікованого COVID та SARS пацієнта [33]

Зразок рентгенографічного зображення для здорової особи подано на рисунку 1.3.

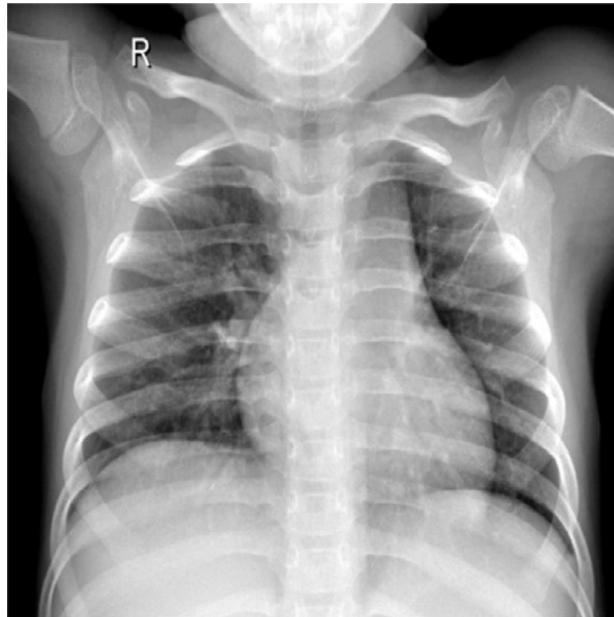


Рисунок 1.3 – Зразок рентгенографічного зображення легенів здорової особи [33]

Зразок рентгенографічного зображення для інфікованого вірусною пневмонією пацієнта подано на рисунку 1.4.

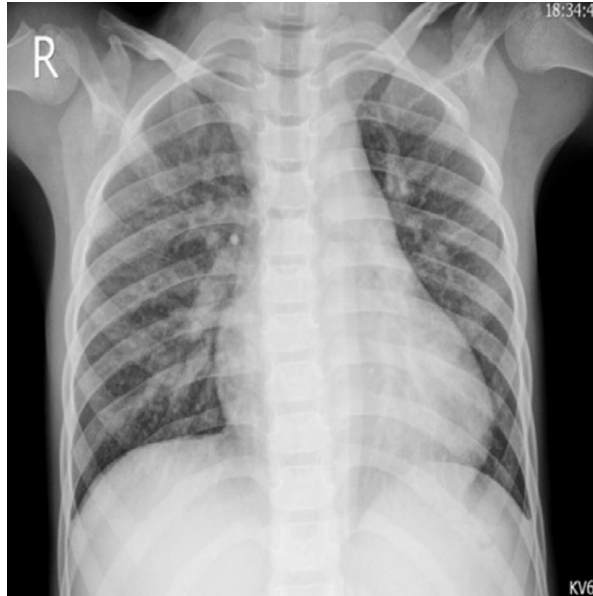


Рисунок 1.4 – Зразок рентгенографічного зображення легенів інфікованого вірусною пневмонією пацієнта [33]

Загальним недоліком досліджень, заснованих на глибокому навчанні, є покладання на здатність узагальнення навчальних вибірок даних. Автори публікують типові зразки зображень із рентгенівської бази COVID та SARS.

Azif і Wenhai [34] запропонували автоматичну діагностичну інформаційну систему COVID та SARS на основі DCNN для обробки рентгенівських зображень. Рентгенівські зображення обробляються за допомогою моделі Inception-V3 на основі DCNN, яка використовує процеси навчання та передачі без попередньої обробки. Такий підхід забезпечує точність класифікації на рівні 96%.

Тогачар, Ерген і Комерт опублікували в [35] інформацію про використання моделей глибокого навчання для виявлення випадків зараження COVID та SARS за допомогою рентгенівських зображень. Це дослідження описує класи, які реструктуризовані за допомогою техніки нечітких кольорів на етапі первинної обробки. Після цього виконується процес порівняння та об'єднання структурованого та вихідного зображень. На наступному етапі зібрані набори даних використовуються на основі моделі глибокого навчання "MobileNetV2" і

Стисніть сітку. Для класифікації зображень дослідники використовували

метод опорних векторних машин (SVM). За допомогою методу, запропонованого авторами роботи, забезпечується коефіцієнт класифікації 99,27%.

Акар і Коркмаз [36] розробили структуру штучного інтелекту для оцінки перебігу COVID та SARS на основі обробки рентгенівських зображень грудної клітки. Тим часом SqueezeNet було налаштовано для діагностики COVID та SARS за допомогою програми байєсівської оптимізації. Додавання набору даних завершено. Дослідники оголосили про точність класифікації COVID та SARS у 98,3 відсотка.

Група авторів під керівництвом Озтюрка [37] запропонувала автоматизовану систему виявлення COVID та SARS на основі алгоритму глибокого навчання. Розроблено модель «DarkCovidNet» для виявлення COVID та SARS в автоматизованому режимі під час обробки рентгена грудної клітки. Дослідження не надає інформації про початкову обробку рентгенівських знімків. Точність класифікації трьох типів даних становить 87,02%.

Хан [38] описав розроблену CNN архітектуру «CoroNet» для виявлення COVID та SARS на основі аналізу КТ та рентгенівського сканування. Модель заснована на архітектурі «Xception». Попереднє навчання на наборі даних ImageNet. Виходячи з опублікованих результатів використання в реальному світі, запропонована автором архітектура забезпечує 89,6% точності для чотирьох класів і 95% для трьох класів.

У своїй роботі [39] Шарма описує процес створення моделі глибокого навчання, яка виявляє пацієнтів, інфікованих COVID та SARS, на основі аналізу рентгенівських зображень. Для збільшення кількості рентгенівських зображень було проведено 25 різних видів збільшення. Повідомляється про покращення продуктивності порівняно з попередніми дослідженнями. Нарін [40] розробив новий метод діагностики COVID-

19 На підставі рентгенограм грудної клітки. Розроблено моделі глибокого навчання «InceptionV3», «ResNet50» та «Inception-ResNetv2». У дослідженні була проведена бінарна класифікація. Набір даних було перехресно перевірено п'ять разів. Це дозволяє забезпечити середню точність 98% для моделі

«ResNet50».

У дослідженні Сінгха [41] архітектура «VGG16» була налаштована після масштабування зображення та етапів початкової обробки, а також використовувалася модель виділення ознак із КТ-зображень легенів людини. Класифікація проводиться на основі чотирьох класифікаторів. Зокрема, CNN, ELM, Online Sequence ELM і Bagging Ensemble with SVM. Найвищий показник успіху в 95,7% був досягнутий за допомогою Bagging Ensemble з SVM.

На підставі аналізу вищезазначених досліджень можна зробити висновок, що на результати діагностики COVID та SARS принципово впливають якісні та кількісні особливості вхідного набору зображень та архітектури мережі. CNN, трансферне навчання та машинне навчання часто ефективно використовуються для діагностики COVID та SARS. Більшість опублікованих робіт містять описи операцій зміни розміру зображення, кадрування тощо. Відповідно до результатів, опублікованих Лю [42], BiLSTM має кращу продуктивність і точність класифікації, ніж згорткові шари. Тому «БіЛСТМ» є дуже сучасним і забезпечує кращі результати класифікації порівняно з CNN.

## 2 МОДЕЛІ РОЗПІЗНАВАННЯ ВІРУСНИХ ЗАХВОРЮВАНЬ ЛЕГЕНІВ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ

### 2.1 Раннє виявлення COVID та SARS з використанням CNN

Загальним недоліком досліджень, заснованих на глибокому навчанні, є покладання на здатність узагальнення навчальних вибірок даних. Автори публікують типові зразки зображень із рентгенівської бази COVID-19.

Azif і Wenhai [34] запропонували автоматичну діагностичну інформаційну систему COVID-19 на основі DCNN для обробки рентгенівських зображень. Рентгенівські зображення обробляються за допомогою моделі Inception-V3 на основі DCNN, яка використовує процеси навчання та передачі без попередньої обробки. Такий підхід забезпечує точність класифікації на рівні 96%.

Тогачар, Ерген і Комерт опублікували в [35] інформацію про використання моделей глибокого навчання для виявлення випадків зараження COVID-19 за допомогою рентгенівських зображень. Це дослідження описує класи, які реструктуризовані за допомогою техніки нечітких кольорів на етапі первинної обробки. Після цього виконується процес порівняння та об'єднання структурованого та вихідного зображень. На наступному етапі зібрані набори даних використовуються на основі моделі глибокого навчання "MobileNetV2" і

Стисніть сітку. Для класифікації зображень дослідники використовували метод опорних векторних машин (SVM). За допомогою методу, запропонованого авторами роботи, забезпечується коефіцієнт класифікації 99,27%.

Акар і Коркмаз [36] розробили структуру штучного інтелекту для оцінки перебігу COVID-19 на основі обробки рентгенівських зображень грудної клітки. Тим часом SqueezeNet було налаштовано для діагностики COVID-19 за допомогою програми байєсівської оптимізації. Додавання набору даних завершено. Дослідники оголосили про точність класифікації COVID-19 у 98,3 відсотка.

Група авторів під керівництвом Озтюрка [37] запропонувала автоматизовану систему виявлення COVID-19 на основі алгоритму глибокого навчання. Розроблено модель «DarkCovidNet» для виявлення COVID-19 в автоматизованому режимі під час обробки рентгена грудної клітки. Дослідження не надає інформації про початкову обробку рентгенівських знімків. Точність класифікації трьох типів даних становить 87,02%.

Хан [38] описав розроблену CNN архітектуру «CoroNet» для виявлення COVID-19 на основі аналізу КТ та рентгенівського сканування. Модель заснована на архітектурі «Xception». Попереднє навчання на наборі даних ImageNet. Виходячи з опублікованих результатів використання в реальному світі, запропонована автором архітектура забезпечує 89,6% точності для чотирьох класів і 95% для трьох класів.

У своїй роботі [39] Шарма описує процес створення моделі глибокого навчання, яка виявляє пацієнтів, інфікованих COVID-19, на основі аналізу рентгенівських зображень. Для збільшення кількості рентгенівських зображень було проведено 25 різних видів збільшення. Повідомляється про покращення продуктивності порівняно з попередніми дослідженнями. Нарін [40] розробив новий метод діагностики COVID-

19 На підставі рентгенограм грудної клітки. Розроблено моделі глибокого навчання «InceptionV3», «ResNet50» та «Inception-ResNetv2». У дослідженні була проведена бінарна класифікація. Набір даних було перехресно перевірено п'ять разів. Це дозволяє забезпечити середню точність 98% для моделі «ResNet50».

У дослідженні Сінгха [41] архітектура «VGG16» була налаштована після масштабування зображення та етапів початкової обробки, а також використовувалася модель виділення ознак із КТ-зображень легенів людини. Класифікація проводиться на основі чотирьох класифікаторів. Зокрема, CNN, ELM, Online Sequence ELM і Bagging Ensemble with SVM. Найвищий показник успіху в 95,7% був досягнутий за допомогою Bagging Ensemble з SVM.

На підставі аналізу вищезазначених досліджень можна зробити висновок, що на результати діагностики COVID-19 принципово впливають якісні та



кількісні особливості вхідного набору зображень та архітектури мережі. CNN, трансферне навчання та машинне навчання часто ефективно використовуються для діагностики COVID-19. Більшість опублікованих робіт містять описи операцій зміни розміру зображення, кадрування тощо. Відповідно до результатів, опублікованих Лю [42], BiLSTM має кращу продуктивність і точність класифікації, ніж згорткові шари. Тому «БіЛСТМ» є дуже сучасним і забезпечує кращі результати класифікації порівняно з CNN.

Модель має наскрізну структуру навчання. Це забезпечує автоматичне виявлення COVID та SARS на основі автоматизованого аналізу рентгенівських знімків грудної клітки без використання методів ручного виділення ознак. Експерти-радіологи можуть використовувати швидкі та стабільні інформаційні системи для підтримки прийняття рішень. Це зменшує навантаження на лікаря-рентгенолога та запобігає постановці помилкових діагнозів.

Хоча метод, розглянутий у [44], наразі є успішним, у майбутньому потрібні більш масштабні дослідження щодо глибокого навчання та методів на основі CNN для виявлення COVID та SARS. Необхідно підвищити успішність виявлення випадків COVID та ГРВІ шляхом збільшення кількості наборів даних опитування. Generative Adversarial Network (GAN) у поєднанні з Deep Neural Network (DNN) потребують подальшого розвитку. Більш детальні алгоритми сегментації легенів на основі CNN також потребують дослідження.

Таблиця 2.1 – Порівняння методів аналітичного опрацювання медичних зображень [33]

зображень [33]		
Автор	Метод	Точність, %
1	2	3
Укар та Коркмаз [36]	Bayes-SqueezeNet	98.30
Озтарк та ін. [37]	DarkCovidNet	87.02
1	2	3
Ванг та Вонг [43]	БЬ	92.30
Гупта та ін. [44]	Ш1аСоу№1-19	99.08
Ху та ін. [45]	Кєв№1 + Фокус уваги	86.70
Сеті та Бехера [46]	КєвКє150 + БУМ	95.40
Хемдан, Шоуман та Карар [47]	УОП19	90.00
Рахімзадех та Аттар [48]	Хєєр1ноп + Кєв№150У2	91.40
Аслан та ін. [33]	Гібридний метод	98.70
Афшар та ін. [49]	Капсульні мережі	95.70
Ноур, Кьомерт та Полат [31]	Машинне БЬ	98.97
Апостопоулос та Мпєсьяна [50]	Передача навчання	93.48
Кхан, Шан та Бхат [38]	Передача навчання	95.00
Язіф та Венхай [34]	Передача навчання	96.00
Фарук та Хафіз[51]	Передача навчання	96.20
Чоударі та ін. [32]	Передача навчання	97.94
Нарин, Кая та Памук [40]	Передача навчання	98.00

## 2.2 Згорткові нейромережі

На даний момент ключова архітектура успіху CNN побудована на глибокому навчанні. Зазначена архітектура нейронної мережі має різні застосування в системах комп'ютерного зору та може використовуватися для розпізнавання різних типів об'єктів у вхідних зображеннях. Ключовою властивістю CNN є те, що він вимагає менше попередньої обробки вхідних даних для виявлення сутностей і шаблонів у вхідних зображеннях, ніж інші алгоритми класифікації. Щоб витягти сутності та функції, інтегровані в зображення, CNN використовують дві зосереджені операції, зокрема злиття та згортання. На першому етапі витягнуті сутності та функції використовуються для навчання моделі. На другому етапі вони використовуються для класифікації зображень. Розглянемо детальний опис кожного шару [54].

Згорткові шари діють як вхідні фільтри, які обробляють одну або кілька піксельних областей зображення. Значення точок формуються між областю перегляду зображення і вагами, встановленими в шарі. рівень активації. Завдяки роботі згорткового шару повертається дуже маленька матриця, значно менша за вихідне зображення. Рівень активації робить характеристики вхідних даних, отриманих від згорткового шару, нелінійними. Це допомагає тренувати нейронну мережу завдяки зворотному поширенню. Функції активації можуть

використовувати Sigmoid, Hyperbolic Tangent, "ReLU" тощо.

Комбінований шар. У цьому шарі розмір матриці, отриманої з рівня активації, зменшується. Це також робиться шляхом вибору максимального або мінімального значення в кожному наборі значень. Зменшення розмірності в комбінованому шарі прискорює процес навчання. Ви можете використовувати кілька шарів злиття подібно до рівнів активації. За допомогою функції шару рендерингу. При цьому спільна функція може мати максимальне значення, мінімальне значення тощо.

Повністю підключений шар отримує одновимірний масив значень з попереднього шару. Результати формуються як набір ймовірностей, що відповідають окремим міткам, прикріпленим до зображення. Мітка, яка отримала найвище значення ймовірності, відповідає контрольному зображенню, визначеному моделлю.

### **2.3 Гібридна модель архітектури CNN**

Новий набір даних відрізняється від попереднього вмісту навчальних даних CNN. У той же час основні функції схожі. При передачі параметрів попередньо навченої моделі нова модель CNN може отримати потужну здатність вилучення ознак і зменшити навчальні обчислення та пам'ять нової моделі. Архітектура передачі навчання показана на рисунку 2.1.

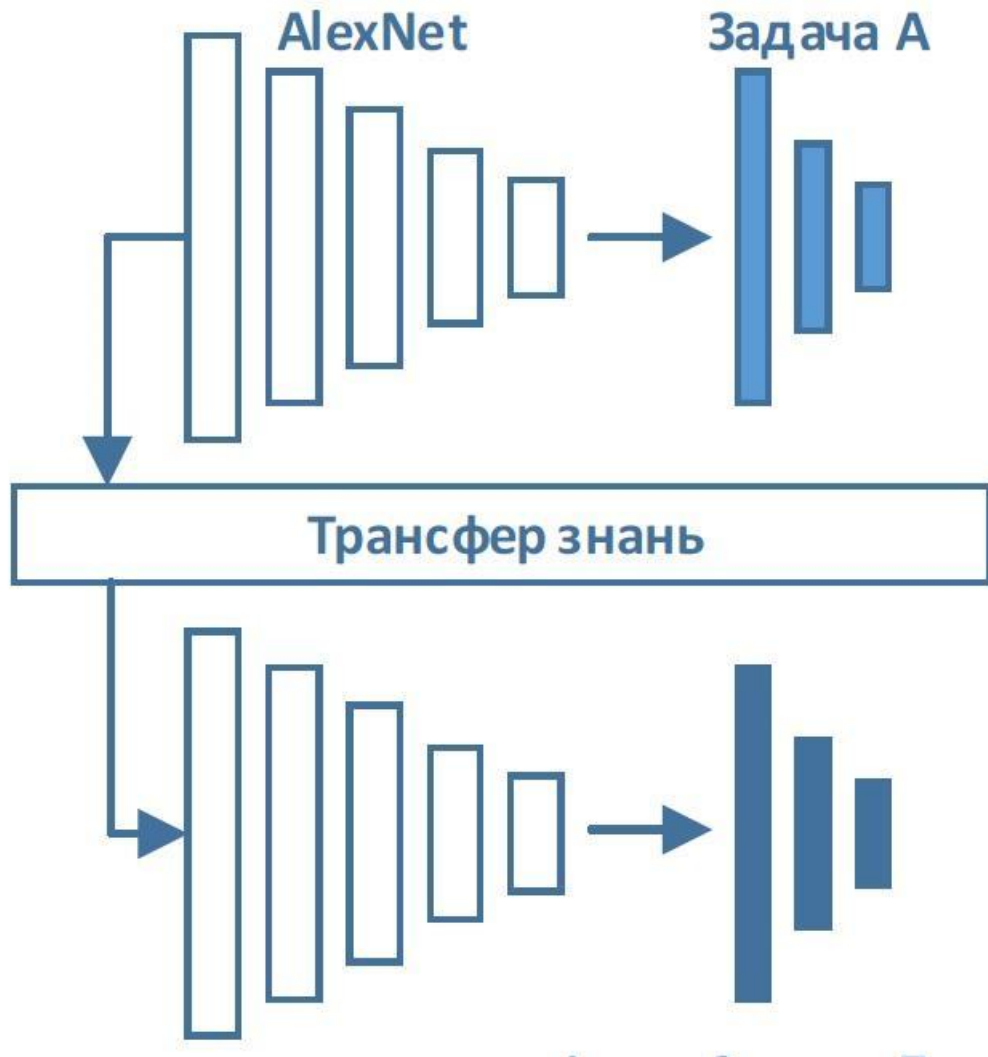


Рисунок 2.1 – Приклад трансфера навчальних знань

У міру формування нових завдань додаються нові шари, останні три шари попередньо навченої моделі CNN видаляються (див. рис. 2.2), а модифікована CNN використовується для нового завдання класифікації.

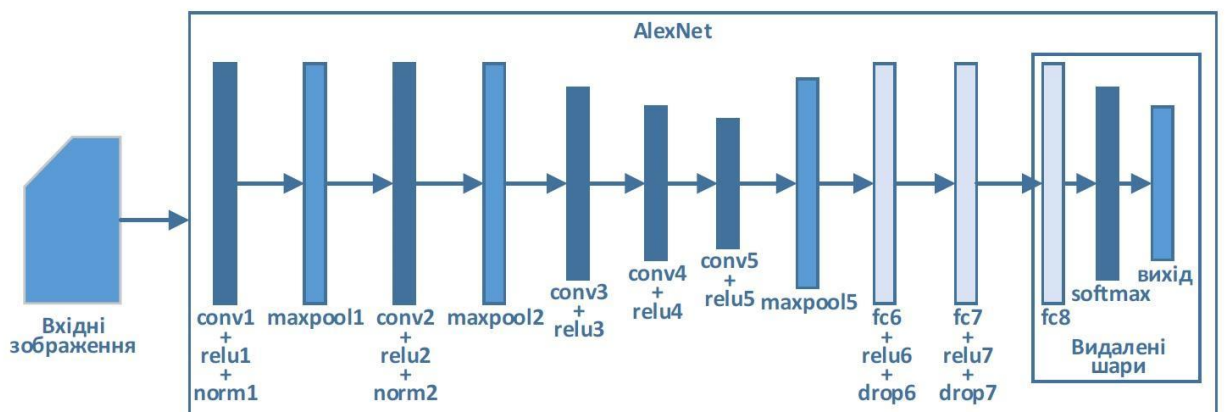


Рисунок 2.2 – Архітектура AlexNet

Друга архітектура для виявлення COVID і SARS повністю охоплює першу архітектуру. Їхні згорткові структури однакові. Гібридна структура відрізняється від попередньої архітектури наявністю шарів BiLSTM. Як показано на рисунку 2.3, шари Flatten і BiLSTM додаються до попередньої архітектури.

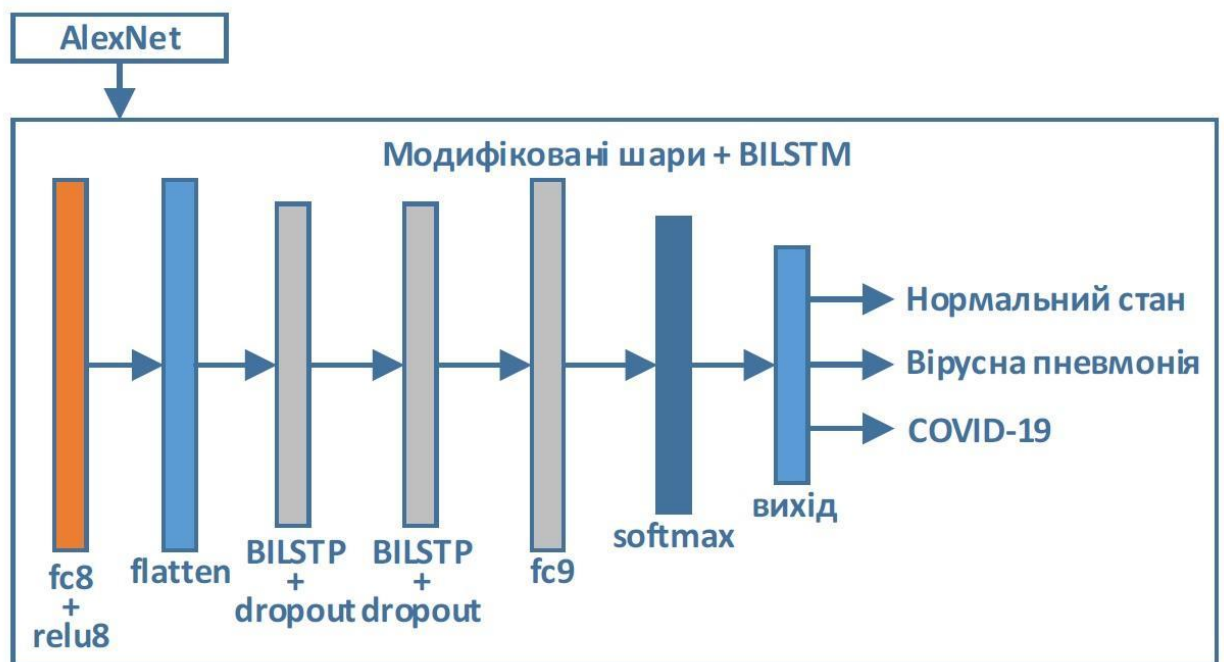


Рисунок 2.3 – Гібридна архітектура mAlexNet-BiLSTM

Зведені шари використовуються для перетворення даних в одновимірні масиви під час підготовки до введення на рівень BiLSTM. BiLSTM і LSTM сформовані на основі архітектури RNN для обробки даних послідовності. На відміну від традиційних алгоритмів нейронної мережі, RNN забезпечують зворотний зв'язок між вхідними та вихідними даними. Таким чином, RNN підходять для послідовних і тимчасових даних. Оскільки LSTM вирішували проблему зникаючого градієнта RNN, він поступово став популярним. BiLSTM перевіряє зв'язок між виходом і напрямком виходу під час зберігання історичної інформації. Таким чином, він забезпечує більш успішні результати для послідовного в порівнянні з LSTM.

Шари BiLSTM добре підходять для аналітичної обробки послідовних даних. Оскільки функції витягуються з рентгенівських зображень, найкращим буде використання згорткових архітектур. Оновлена архітектура включає згорткові шари та BiLSTM (див. рис. 2.3). Значення параметрів гібридної архітектури mAlexNet-BiLSTM наведені в таблиці 2.2

Таблиця 2.2 – Характеристики шарів mAlexNet

Назва шару	Розмір	Розмір фільтра	Вкладення	Прокладка	Вихідний канал	Функція активації
conv1	55 x 55	11 x 11	4	0	96	relu
maxpool1	27 x 27	3 x 3	2	0	96	—
conv2	27 x 27	5 x 5	1	2	256	relu
maxpool2	13 x 13	3 x 3	2	0	256	—
conv3	13 x 13	3 x 3	1	1	384	relu
conv4	13 x 13	3 x 3	1	1	384	relu
conv5	13 x 13	3 x 3	1	1	256	relu
maxpool5	6 x 6	3 x 3	2	0	256	—
1	2	3	4	5	6	7
fc6	—	—	—	—	4096	relu
fc7	—	—	—	—	4096	relu
fc8	-	—	—	—	25	relu
fc9	-	—	—	—	2	softmax

Навчальні характеристики гібридної архітектури mAlexNet-BiLSTM подано в таблиці 2.2.

Таблиця 2.3 – Параметри тренування mAlexNet

Оптимізатор	Макс. епох	Мін. партія	Початкова швидкість навчання ( $\alpha$ )	Імпульс ( $\gamma$ )
SGDM	100	60	0.001	0.95

Використовуються два послідовних шари BiLSTM, зокрема BiLSTM-1 і BiLSTM-2. Тимчасові особливості, виділені за допомогою цих шарів, надаються як вхідні дані для наступного підключеного шару fc9 і шару Softmax. У розглянутій архітектурі кількість нейронів у прихованому шарі, функція активації та інші параметри вибиралися експериментально (див. табл. 2.4).

Таблиця 2.4 – Гіперпараметри шарів гібридної архітектури

BiLSTM-1			BiLSTM-2			fc9	
Кількість прихованих шарів	Функція активації		Кількість прихованих шарів	Функція активації		Вихідний розмір	State функція активації
	State	Gate		State	Gate		
125	<i>tanh</i>	<i>sigmoid</i>	100	<i>tanh</i>	<i>sigmoid</i>	3	<i>tanh</i>

BiLSTM-1			BiLSTM-2			fc9	
Кількість прихованих шарів	Функція активації		Кількість прихованих шарів	Функція активації		Вихідний розмір	State функція активації
	State	Gate		State	Gate		
125	<i>tanh</i>	<i>sigmoid</i>	100	<i>tanh</i>	<i>sigmoid</i>	3	<i>tanh</i>

Таблиця 2.5 - Тренувальні параметри гібридної архітектури

Оптимізатор	Коефіцієнт градієнтного розпаду ( $\alpha$ )	Квадратичний коефіцієнт розпаду градієнта ( $\beta$ )	Макс. епох	Мін. розмір партії	Початкова швидкість навчання( $\eta$ )	Епсилон ( $\epsilon$ )
Adam	0.9	0.999	200	512	0.001	10-8

Під час фази навчання алгоритм оптимізації Адама використовується для зменшення помилки на кожній ітерації. Алгоритм оптимізації Адама — це адаптивний алгоритм швидкого навчання, спеціально розроблений для глибоких нейронних мереж. Алгоритм Адама перевершує інші методи оптимізації нейронної мережі завдяки відносно низьким вимогам до пам'яті [53]. Алгоритм Адама – це метод адаптивного збільшення швидкості навчання. Розраховує індивідуальні характеристики темпів навчання за різними критеріями та параметрами. Адаптивна швидкість навчання – це коригування швидкості навчання нейронної мережі в процесі навчання за рахунок зниження швидкості навчання за заздалегідь встановленим графіком. Алгоритм Адама заснований на методі оптимізації SGDM і RMSProp. Він використовує оновлення параметрів, подібно до RMSProp. Але на відміну від RMSProp, він включає затримку імпульсу. Алгоритм отримав назву «Адам» через оцінку адаптивних моментів. Він використовує передбачення першого та другого моментів градієнта для адаптації швидкості навчання кожної ваги нейронної мережі. У цьому випадку використовуються прогнозовані значення для першого і другого ступенів періоду оновлення.

## 2.4 Капсульні мережі

Архітектура “CapsNet” є альтернативою CNN, запропонованій Хінтоном [55]. Щоб зменшити дискретність, архітектури CNN містять рівні об'єднання.



Об'єднання функцій застосовується до карт об'єктів, щоб підтвердити розпізнавання CNN схожих об'єктів на зображеннях різного масштабу. У результаті вимоги до пам'яті моделі зменшуються через меншу кількість семплів. Це в свою чергу призводить до формування просторової інваріантності СНС. Просторова інваріантність є одним з головних недоліків CNN. Іншими проблемами CNN є нездатність розпізнавати спотворення текстури, розрізняти пози та форми цілих зображень або їх частин. CNN є інваріантними через операції конкатенації, і не є еквіваріантним. Тому їм бракує еквівалентності. Через зменшення дискретизації CNN втрачають деякі особливості зображення. Щоб компенсувати втрату, необхідно гарантувати розмір навчальних вибірок. Оскільки CNN мають велику глибину архітектури, вони характеризуються збільшеним часом навчання порівняно з CapsNets. CapsNet, у свою чергу, має широку архітектуру, але обмежену кількість параметрів. На даний момент існує кілька відомих підходів до проектування CapsNet. Зокрема, векторні капсули на основі динамічної маршрутизації, автокодерів перетворення [56] і формування матриці на основі маршрутизації максимізації очікування [57].

В інкапсульованому CapsNet вектор формується для представлення ймовірності існування об'єкта. При цьому орієнтація використовується для опису його властивостей. Кожен розмір капсули можна використовувати для опису окремого атрибута. Зокрема розмір, положення, текстура, деформація, швидкість, тіні тощо. Нелінійні функції або функції стиснення використовуються для полегшення класифікації різних капсул під час навчання. Використовується для активації на рівні індивідуальної капсули. Функція стиснення обмежує довжину капсули менше ніж 1. Функції стиснення можна ефективно використовувати для класифікації існуючих наборів об'єктів. Однорівневі обмеження в CapsNet схожі на рівні, доступні в основних нейронних мережах. У нижній кришці нижнього шару багато капсул. Вони невеликі за розміром. У кришках вищого рівня капсул менше, і вони набагато більші за розміром. Така організація нейронної мережі використовується тому, що на нижньому рівні одна капсула представляє невелику вхідну область. Завдяки великій кількості таких капсул легко формуються просторові

співвідношення. На верхньому рівні ширший список вимірювань використовується для зберігання інформації про розташування об'єктів. Тому, коли капсули переміщуються від нижчих рівнів до вищих, їх розмір збільшується. Після завершення процедури збільшення розміру можна встановити співвідношення між окремими капсулами та кришками. Тим часом капсули вищого рівня використовуються як зважені суми капсул нижчого рівня. Динамічна маршрутизація, яка використовується на цьому етапі, допомагає перетворити одну капсулу нижчого рівня в капсулу вищого рівня [58].

CapsNet використовує кілька окремих шарів капсули. Архітектура CapsNet, показана на рисунку 2.4, містить згортковий рівень ReLU. Після зазначеного шару розміщується шар первинних капсул, який був стиснутий і змінений у розмірі поверх згорткового шару. Далі йде шар X-rayCaps з капсулами для ідентифікації трьох типів рентгенівських зображень. Згорткові шари використовуються для інтерпретації інтенсивності пікселів у роботі детектора локальних ознак. Результати передаються в первинні капсули. Згорткові шари з кроком понад 2 використовуються для зменшення розмірності порівняно з шарами CNN із максимальним об'єднанням. Вихід X-rayCaps використовується для класифікації вхідного зображення.

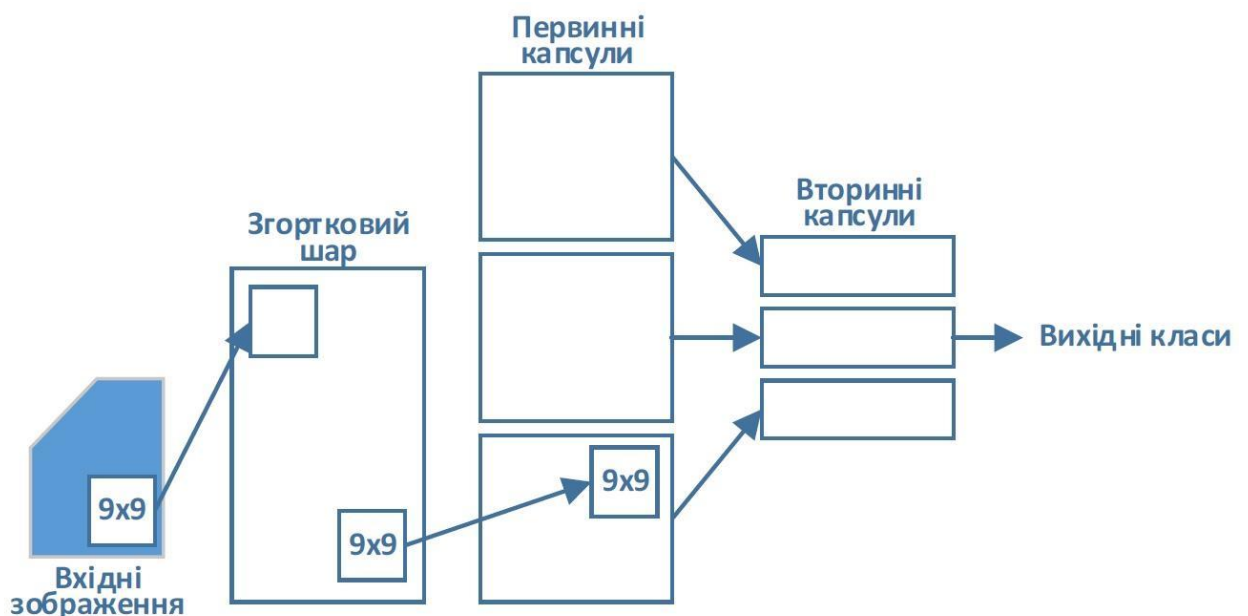


Рисунок 2.4 – Архітектура CapsNet для виявлення COVID та SARS

Кожна окрема капсула в шарі X-rayCaps з'єднана з кожною капсулою в основному шарі (див. рис. 2.5). Підхід протокольної маршрутизації може забезпечити кращі показники навчання, ніж операції, засновані на максимальному об'єднанні, утвореному маршрутизацією.

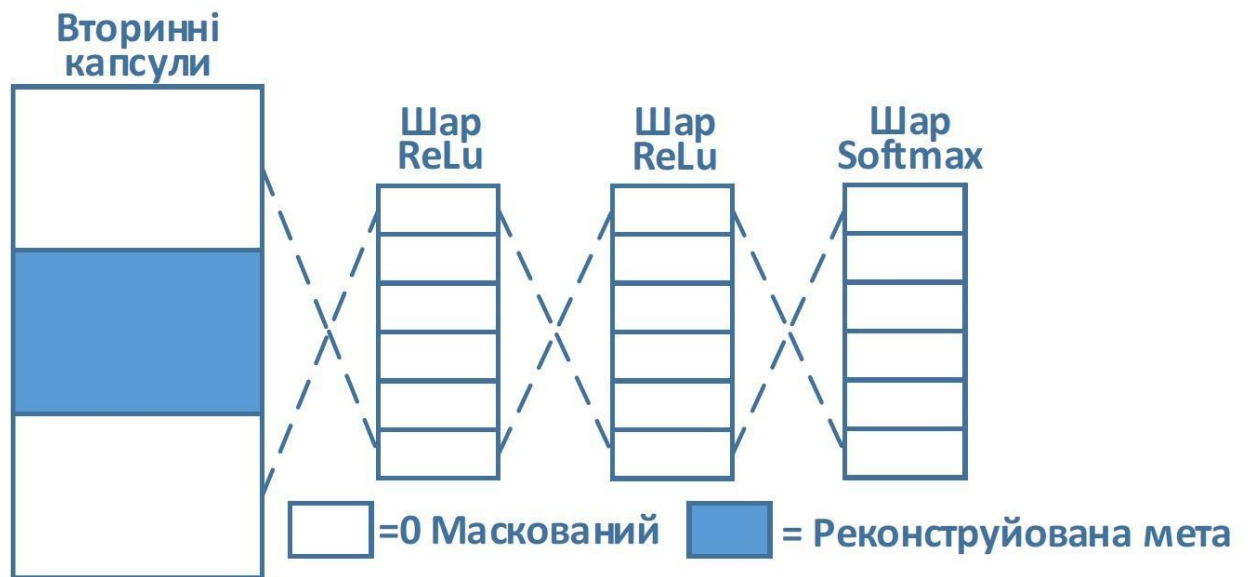


Рисунок 2.5 – Запропонована Сабоуром [56] архітектура капсульної мережі для виявлення COVID та SARS

У динамічній маршрутизації процес зворотного зв'язку додає підтримку для капсули, яка найбільше відповідає вихідному сигналу відповідно до методу відповідності. Це, у свою чергу, посилює його підтримку. Слід зазначити, що функції активації використовуються для апроксимації нелінійних з'єднань у глибоких мережах. Вони реалізовані як базові математичні функції, зазвичай для скалярних величин. У мережі CapsNet для нормалізації векторів використовується спеціальна функція сквошу, нелінійна функція активації.

## 2.5 Неймерева модель VGG-CapsNet

Дослідники використовували моделі нейронної мережі CNN і CapsNet для виявлення COVID і SARS відповідно. У [14] запропоновано модель, у якій

структура нейронної мережі CapsNet розширена мережею VGG16 для формування моделі VGG-CapsNet для отримання більш детальної інформації на рентгеновських зображеннях для діагностики COVID та SARS. Мережа VGG16 використовується для формування базової карти об'єктів. Потім карти функцій вводяться в модель CapsNet для виконання процесу класифікації.

Рівень виділення ознак базового рівня CapsNet формується на основі згорткового шару. 256 фільтрів  $9 \times 9$  із кроком 1, розміщених у вказаних згорткових шарах. Він використовує активацію функції ReLU для подальшої передачі обчисленої функції на базовий рівень інкапсуляції. У досліджуваній архітектурі було змінено рівень функціонального розподілу.

Архітектура капсульної мережі VGG-CapsNet, показана на рисунку 2.6, розділена на два компоненти, а саме VGG16 і CapsNet.

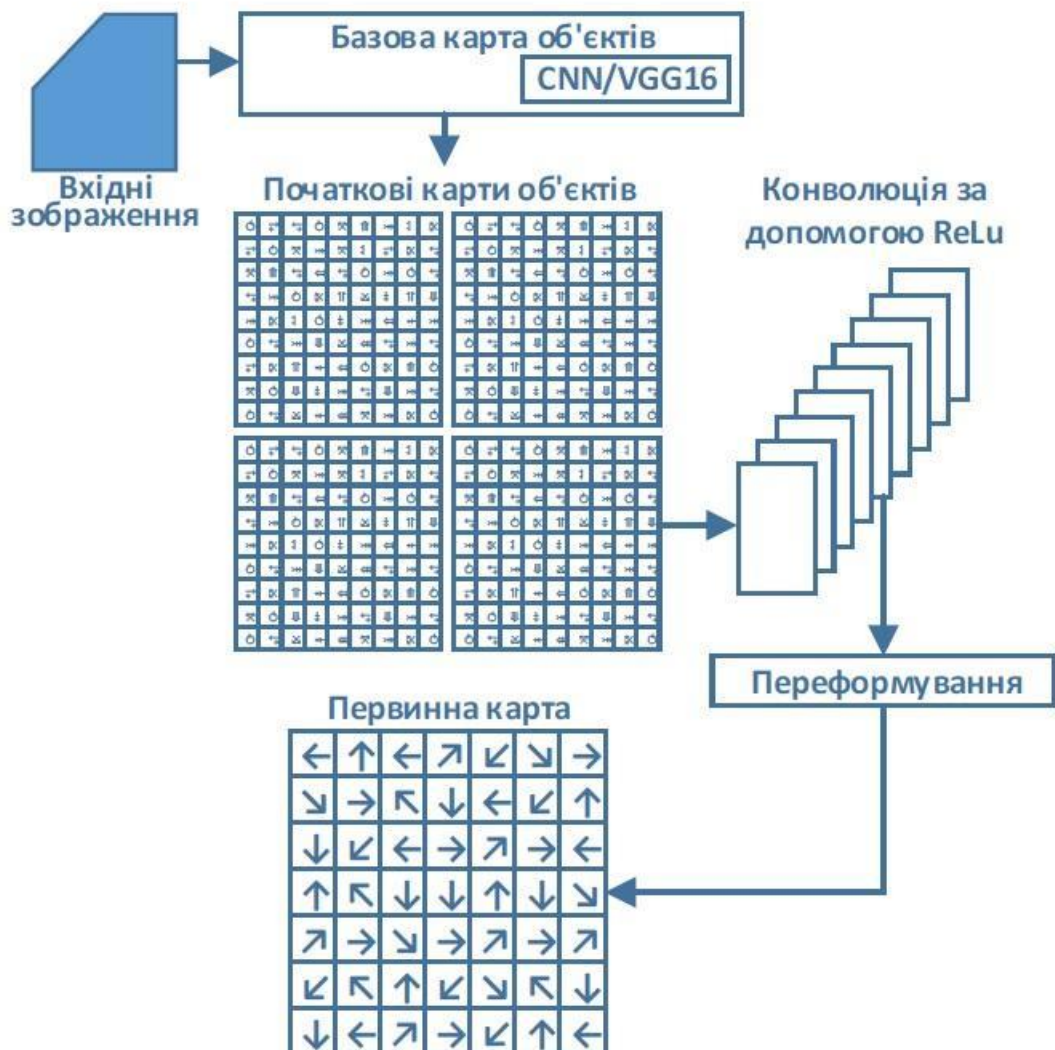


Рисунок 2.6 – Базова архітектура моделі VGG- CapsNet для виявлення COVID та SARS

Він сформований на основі VGG16 для обчислення початкового графіка функції та реалізації архітектури CapsNet на рівнях нижче. На першому етапі рентгенівські зображення надходять на модель VGG16, яка пройшла попередню процедуру підготовки. Згорткові шари використовуються для формування початкової карти ознак. На наступному кроці початкова карта функцій CapsNet передається для виконання класифікації.

У досліджуваній архітектурі початковий детектор ознак було замінено. Це робиться шаром "block5\_pool" на основі моделі VGG16.

Розглянута модифікація реалізує метод навчального переносу. У той же час метою є передача CapsNet можливості представлення низькорівневих функцій моделі VGG16. Це дасть CapsNet можливість розпізнавати функції низького рівня, подібні до VGG16 (див. рис. 2.7).

Архітектура VGG16 була вперше запропонована в 2014 році Симоніаном Зіссермана. Модель VGG16 сформована на основі тринадцяти згорткових шарів. Вони також включають максимальний рівень уніфікації. Також включає два повністю зв'язані шари та класифікатор Softmax. Щоб отримати більш складні функції з меншим споживанням ресурсів у цій мережі, більше ядро замінено декількома окремими фільтрами «3 × 3».

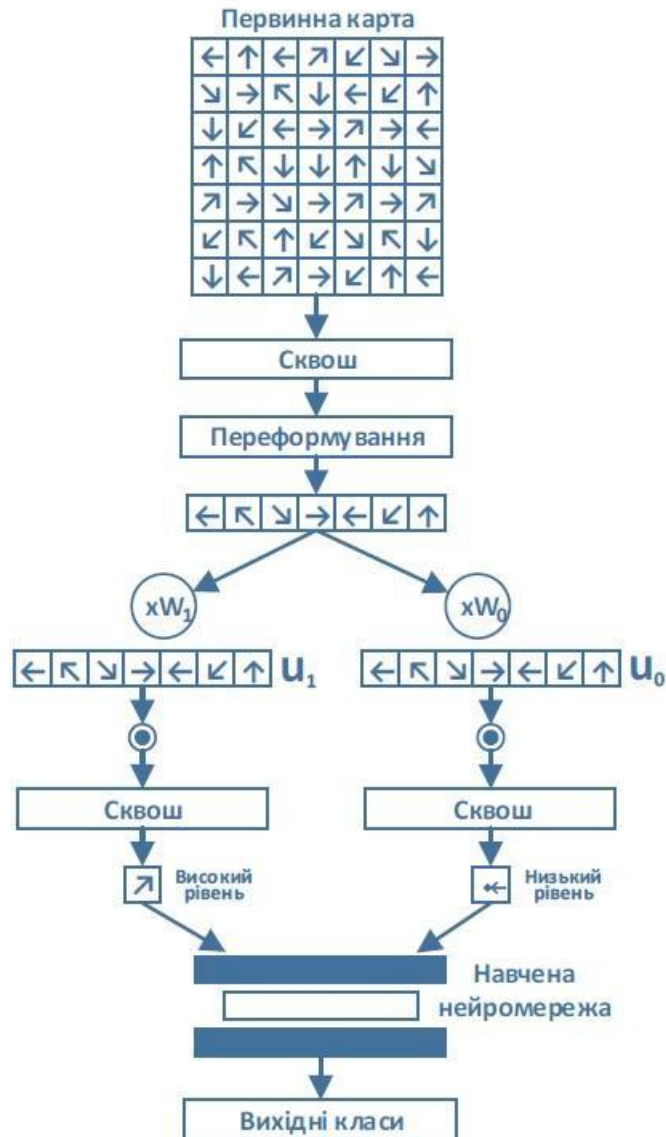


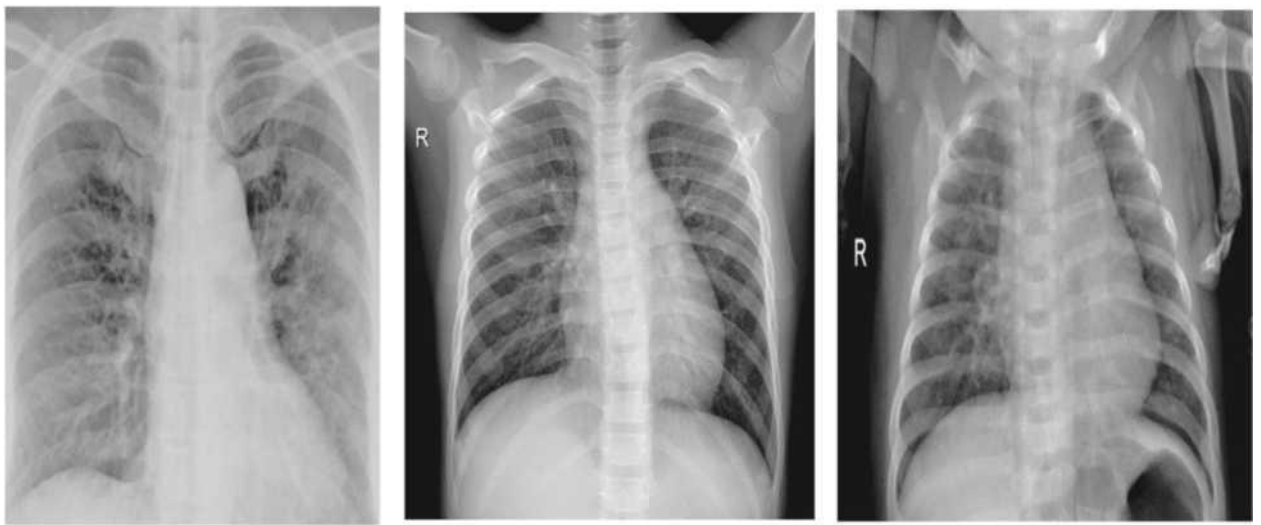
Рисунок 2.7 – Узагальнююча складова архітектури моделі VGG- CapsNet для виявлення COVID та SARS

## 2.6 Вибір датасету

У дослідженнях можна використовувати набір рентгенограм грудної клітки [59]. Тестовий набір містить понад 200 зображень регенерації хворих на COVID та ГРВІ, понад 1300 зображень пневмонії та понад 1300 зображень здорових громадян. Приклади зображень для кожної категорії показані на рисунку 2.8.

Набір даних дослідження розділений на три частини. Зокрема підмножини навчання, тестування та перевірки. Досліджувані набори даних є

незбалансованими. Тому для оцінювання використовується однакова кількість зображень із кожної категорії. Близько 715 зображень було випадковим чином відібрано для підготовки перевірки та тестування наборів даних для багатокласової класифікації для виявлення COVID та SARS. При виявленні COVID та ГРВІ використовували бінарну класифікацію. При цьому звичайні рентгенівські знімки та зображення вірусної пневмонії об'єднані в одне – «не-COVID19».



(A)

(B)

(C)

Рисунок 2.8 – Зразки зображень з обраного датасету (A – COVID, Б – нормальне, С – SARS)

## 3 ОБЧИСЛЮВАЛЬНИЙ ЕКСПЕРИМЕНТ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ ДЛЯ ЗАДАЧ РОЗПІЗНАВАННЯ МЕДИЧНИХ ЗОБРАЖЕНЬ

### 3.1 Обчислювальний експеримент і результати моделювання

Метою обчислювальних експериментів є розвиток та оцінка здатності двох класифікаторів до узагальнення. Особливо моделі CNN і CapsNet. Ефективність кожної моделі буде оцінюватися з точки зору точності класифікації, запам'ятовування та F-оцінки, які є ключовими показниками для вимірювання продуктивності навченої моделі [60].

Усі зображення використовуються для розрахунку показників ефективності. Досліджуваний набір зображень складається з підмножини навчання, підмножини перевірки та підмножини тестування.

Точність:

$$\text{Precision} = T^P / (T^P + F^P) \quad (3.1)$$

Повторюваність:

$$\text{recall} = T^P / (T^P + F^N) \quad (3.2)$$

F-оцінка:

$$F\text{-score} = (2 \times \text{recall} \times \text{precision}) / (\text{recall} + \text{precision}) \quad (3.3)$$

де:

$T^N$  – негативне – коли модель виявлення COVID та SARS спрогнозувала помилку а фактичне значення хибне.

$F^P$  – хибнопозитивне – це помилка першого типу, коли а модель



виявлення COVID та SARS спрогнозувала істину але фактичне значення хибне.

$T^P$  – позитивне – коли модель ідентифікації COVID та SARS передбачила істину і фактичне значення позитивне.

$F^N$  – хибнонегативне – це помилка другого типу, коли фактичне значення позитивне, а модель виявлення COVID та SARS передбачала хибне.

Для візуалізації криві робочих характеристик приймача ROC для обох моделей представлені у двох вимірах. Результати розрахунків і вибір класифікаторів формуються на основі показників ефективності. Метрична площа під кривою ROC є AUC. Зазвичай він обчислюється як одне скалярне значення. AUC – частка площі еталонного одиничного квадрата. Значення знаходиться в діапазоні «0-1,0». Модель вважається відмінною, якщо значення AUC перевищує 0,95. У задачі ідентифікації COVID та SARS розраховуються показники макро- та мікросереднього. Макросередній показник незалежно визначає ефективність кожної категорії. Потім розраховується середнє значення з урахуванням усіх категорій. Вимірювання мікросереднього значення об'єднує рентгенограми грудної клітки в усіх категоріях для обчислення середнього значення.

### **3.2 Модель CNN-CapsNet для виявлення COVID та SARS**

Архітектура CNN-CapsNet формується на основі двох частин, а саме CNN і CapsNet. Спочатку перенесіть рентгенівське зображення на модель CNN. Згорткові шари використовуються для отримання початкових карт функцій. Потім карти функцій завантажуються в модель CapsNet для виконання остаточної класифікації. Модель CNN використовує архітектуру CNN-CapsNet, сформовану на основі чотирьох згорткових шарів. Кожен згортковий рівень використовує 64, 64, 128 і 128 ядер відповідно. Це "3×3"?

Розмір кожного фільтра згортки. Для всіх шарів використовується функція активації "ReLU". Після другого шару згортки розташовується шар проміжного з'єднання. Архітектура CapsNet-моделі CNN-CapsNet описана в розділі 2.2. На даний момент розроблено дві окремі моделі. Один для виявлення

випадків зараження COVID і SARS на основі двійкового таксономічного призначення «COVID і SARS» або «не COVID і SARS». Друга модель виконує класифікацію на основі трьох категорій, зокрема «COVID і SARS»,

«Здоров'я» та «Вірусна пневмонія». При початковій швидкості навчання 0,001 модель узгоджується з оптимізатором Адама. Функція граничних втрат використовується для визначення рентгенівських характеристик класу. Його рівняння:

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \alpha(1 - T_k) \max(0, m^+ - \|v_k\| - m^-)^2 \quad 3.4)$$

де  $T_k = 1$ , якщо ознака класу  $k$  існує, а та  $m^+ = 0.1$  з метою підтвердження, що довжина вектора буде залишатися в практичних межах  $m^+ = 0.9$ .

Функція зменшення ваги використовується для чисельної стабільності. Рекомендується встановити початкове значення 0,5. Обидві моделі тренувалися протягом ста епох. Пакет запуску містить тридцять два значення, а тестовий пакет містить одне значення. Рентгенівські зображення "64x64" використовуються для навчання та тестування мережі. Завершіть тренування методом ранньої зупинки, щоб уникнути наслідків перетренованості. Якщо перевірка завершується подальшим покращенням, механізм ранньої зупинки припиняє навчання навчальних зразків. У цьому випадку функція втрат обчислюється після кожної епохи на тестовому наборі. Навчання припиняється, коли втрати підтвердження збільшуються. Це ознака перенавчання. Скасування кроку навчання залежить від параметра терпіння, який відповідає кількості епох очікування покращення втрати перевірки. Для обох моделей параметр допуску дорівнює десяти.

На рисунку 3.1 показано точність навчання та тестування моделі згорткової нейронної мережі CNN-CapsNet для проблеми класифікації двійкових класів.

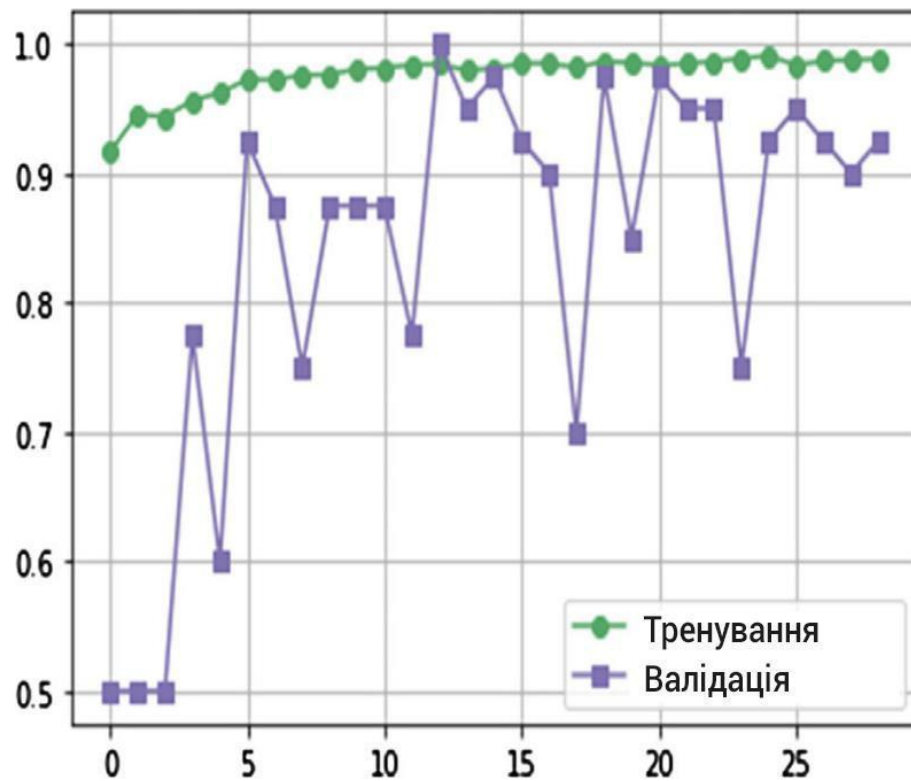


Рисунок 3.1 – Точності моделі CNN-CapsNet для задачі бінарної класифікації

Тим часом використовувався тестовий набір. У перших п'яти ітераціях втрати навчання значно зменшуються. Він стабілізувався після десяти ітерацій. На рисунку 3.2 показано графік втрат класифікації під час навчання та тестування моделі згорткової нейронної мережі CNN-CapsNet для проблеми класифікації двійкових класів. Після десяти епох навчання припинилося після двадцяти дев'яти епох.

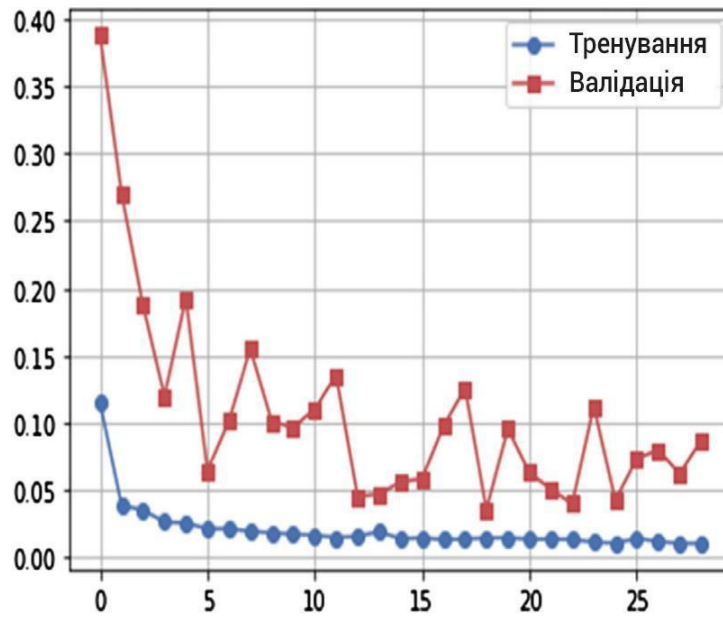


Рисунок 3.2 – Втрати моделі CNN- CapsNet для задачі бінарної класифікації

На рисунку 3.3 зображено точність класифікації набору перевірки згорткової капсульної нейронної мережі CNN-CapsNet для класифікації за трьома класами.

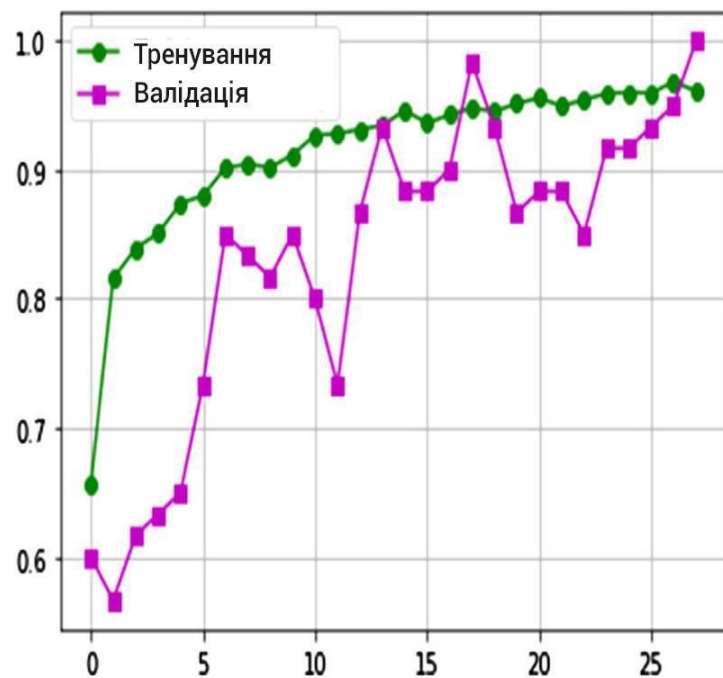


Рисунок 3.3 – Точність моделі CNN-CapsNet для трикласової задачі

На рисунку 3.4 показано навчальний набір і графіки втрат під час

навчання моделі згорткової нейронної мережі Caps (CNN-CapsNet) для класифікації за трьома класами.

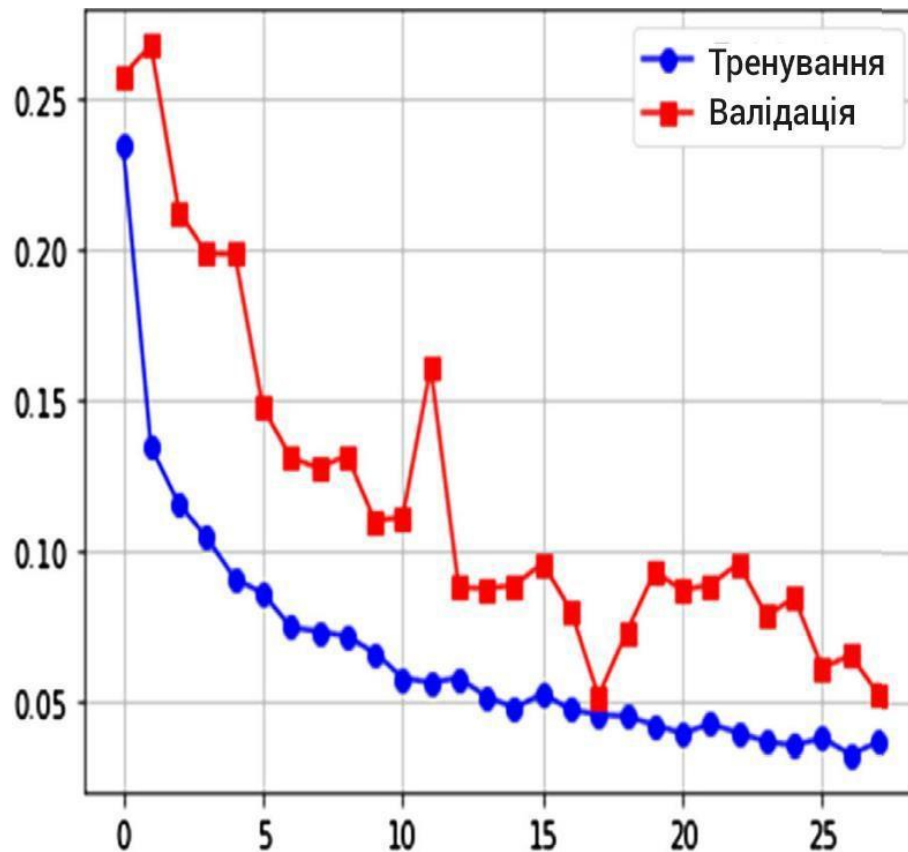


Рисунок 3.4 – Втрати моделі CNN-CapsNet для трикласової задачі

Втрата навчання значно зменшується в перші десять ітерацій і стабілізується після п'ятнадцяти ітерацій. Навчання буде припинено через 29 epoch після 10 epoch.

Матриця плутанини для моделі згорткової капсульної нейронної мережі CNN-CapsNet для задач бінарної класифікації сформована на основі роботи [14], як показано на рисунку 3.5.

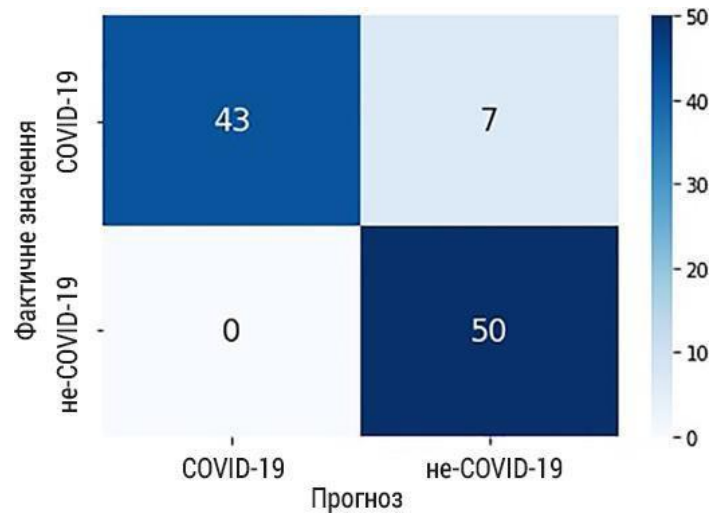


Рисунок 3.5 – Матриця сплутування моделі CNN-CapsNet для двокласової задачі

І навпаки, матриця сплутування моделі згорткової капсульної нейронної мережі CNN-CapsNet для задачі трьох класів, сформованої на основі роботи [14], показана на рисунку 3.6.

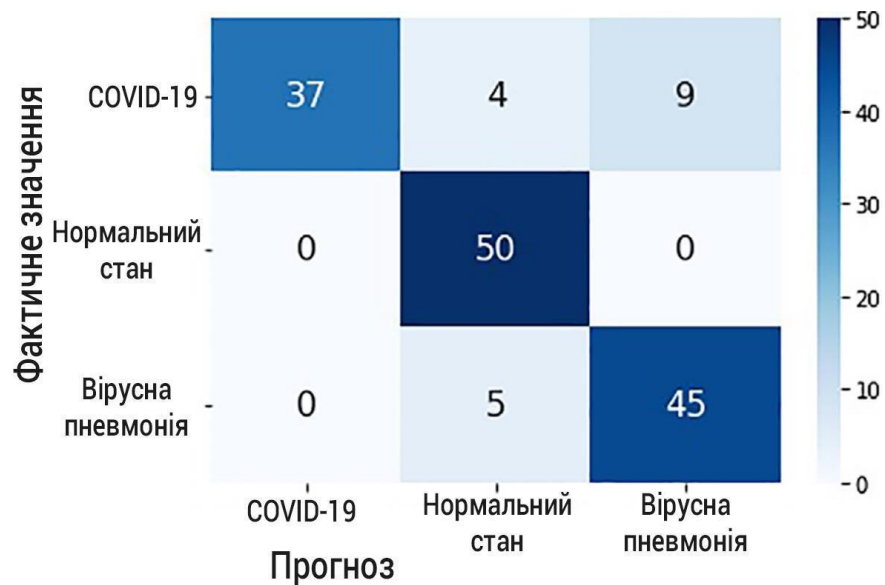


Рисунок 3.6 – Матриця сплутування моделі CNN-CapsNet для трикласової задачі

В таблиці 3.1 подано показники продуктивності сформовані на основі результатів експерименту:

Таблиця 3.1 – Результати ефективності моделей CNN-CapsNet і VGG-CapsNet при вирішенні задачі бінарної класифікації

Клас	CNN-CapsNet			VGG-CapsNet		
	Точність	Запам'ятовуваність	Оцінка F1	Точність	Запам'ятовуваність	Оцінка F1
«COVID та SARS»	1.00	0.90	0.95	0.96	0.98	0.97
«Не-COVID19»	0.91	1.00	0.95	0.98	0.96	0.97
Точність	0.95			0.97		

### 3.3 Модель VGG-CapsNet для виявлення COVID та SARS

Навчена модель CNN VGG-16 використовується для виділення ознак, щоб оцінити ефективність класифікації CapsNet за допомогою згорткових ознак. CNN на основі VGG використовуються для завдань класифікації медичних зображень. Показники точності результатів експериментального впровадження на основі моделей MobileNet і ResNets неможливо порівняти з результатами моделювання на основі VGG16 [14]. Тому модель VGG16 є найкращим вибором. Рентгенівське зображення в масштабі

Відповідно до розміру вхідного шару VGG-16 «299×299» точок. Розроблено дві незалежні моделі VGG-CapsNet. Метод, заснований на двійковій класифікації для виявлення COVID і SARS, зокрема «COVID і SARS» або «не COVID і SARS». Друга модель формує трирівневу класифікацію, а саме: «COVID-19», «нормальна» та «вірусна пневмонія». Ці моделі скомпільовані за допомогою sgd-optimizer. Тим часом початкова швидкість навчання становить 0,00001, імпульс – 0,9, а розпад – 0,001. Двійкова функція втрат крос-ентропії для моделей бінарного класу:

$$L_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N y_i \log(h_w(x_i)) + (1-y_i) \log(1-h_w(x_i)) \quad (3.5)$$

де  $N$  – кількість навчальних вибірок,

$y_i$  – цільова мітка для  $i$ -ї навчальної вибірки,  $x_i$  – вхідні дані для  $i$ -ї навчальної вибірки,

$h_w$  – модель з вагами  $w$ .

Для моделі трьох класів використовується функція втрат «Категоризована крос-ентропія»:

$$L_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}} \quad (3.6)$$

де  $W$  – вагова матриця,

$b$  – член зміщення,

$x_i$  –  $i$ -та навчальна вибірка,

$y_i$  – мітка класу для  $i$ -ї навчальної вибірки,

$N$  – кількість вибірок,

$W_j$  та  $b_{y_i}$  –  $j$ -й і  $y_i$ -й стовпець відповідно.

Обидві моделі тренувалися протягом ста епох. Розмір тестової упаковки становить тридцять дві картинки, а тестової упаковки — дві картинки. Рентгенівські зображення розміром "299x299" використовуються для навчання, перевірки та тестування. Щоб уникнути перенавчання, використовуйте ранню зупинку, щоб припинити процес навчання. Функція втрат обчислюється після кожної епохи в тестовому наборі. Навчання припиняється, коли втрати підтвердження починають збільшуватися. Для обох моделей період терпіння становить десять. Точність моделі візуальної геометричної капсульної мережі VGG-CapsNet для проблеми бінарної класифікації показано на рисунку 3.7.



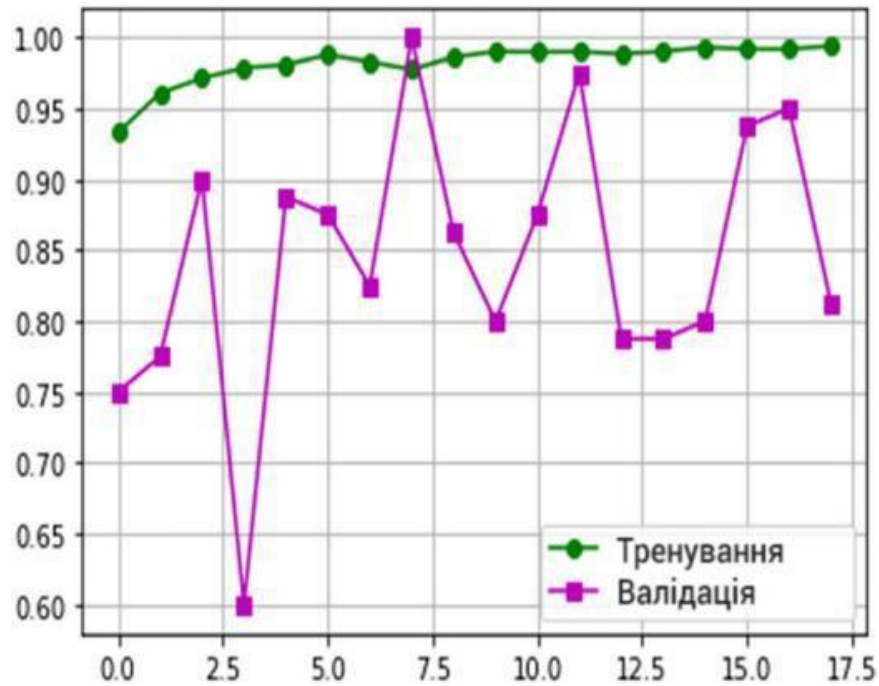


Рисунок 3.7 – Точність моделі VGG- CapsNet для задачі бінарної класифікації

Графік втрат моделі капсульної мережі візуальної геометрії VGG-CapsNet для бінарної класифікаційної задачі подано на рисунку 3.8.

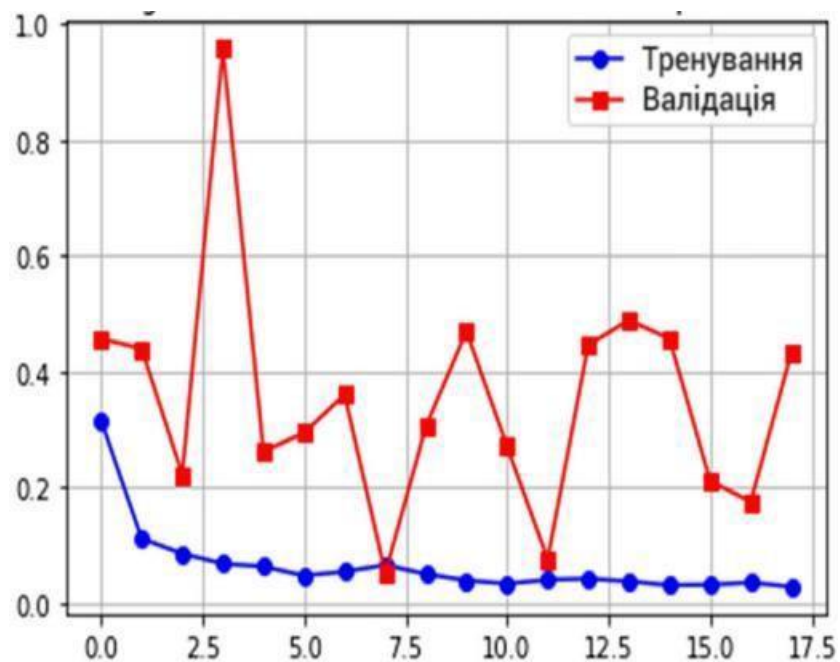


Рисунок 3.8 – Втрати моделі VGG- CapsNet для задачі бінарної класифікації

Втрата навчання значно зменшується під час перших п'яти ітерацій і стабілізується після десяти ітерацій. Після очікування 10 епох навчання повністю припиняється після 17 епох. CNN-CapsNet — це згорточна капсульна нейронна мережа, а VGG-CapsNet — групова капсульна візуальна геометрична мережа.

На рисунку 3.9 показано точність візуальної моделі геометричної капсульної мережі VGG-CapsNet для задачі класифікації на три класи.

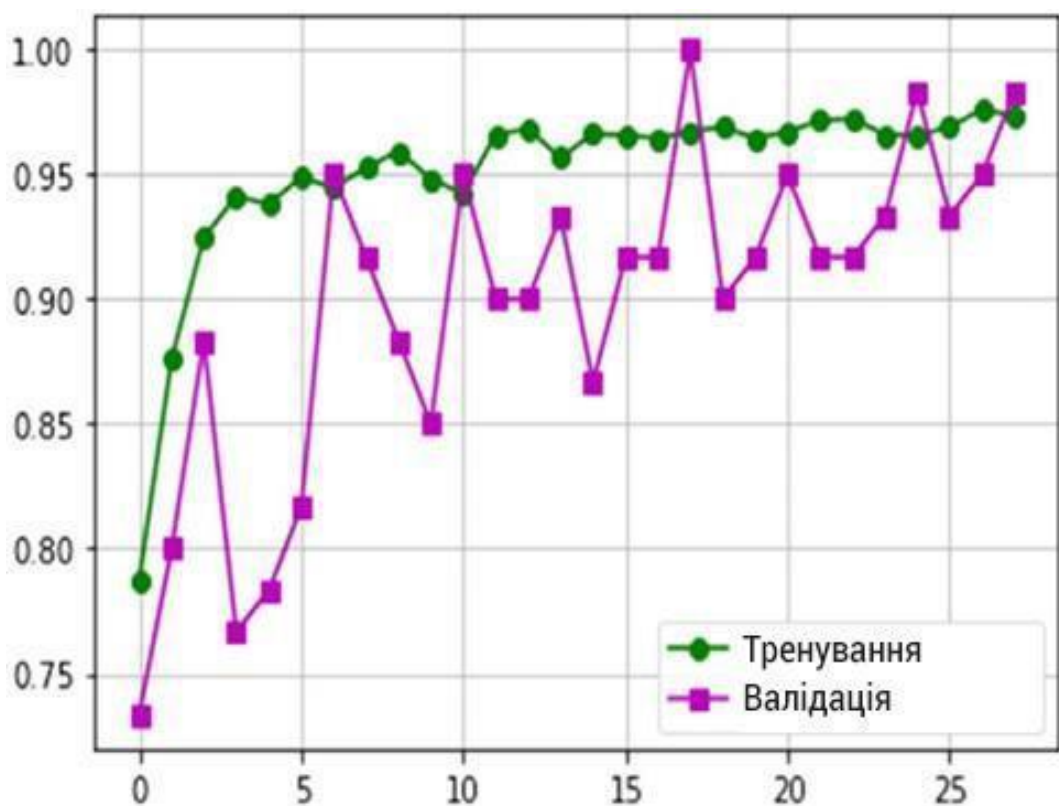


Рисунок 3.9 – Точність моделі VGG- CapsNet для задачі класифікації на три класи

Графік втрат моделі капсульної мережі візуальної геометрії VGG-CapsNet для задачі класифікації на три класи подано на рисунку 3.10.

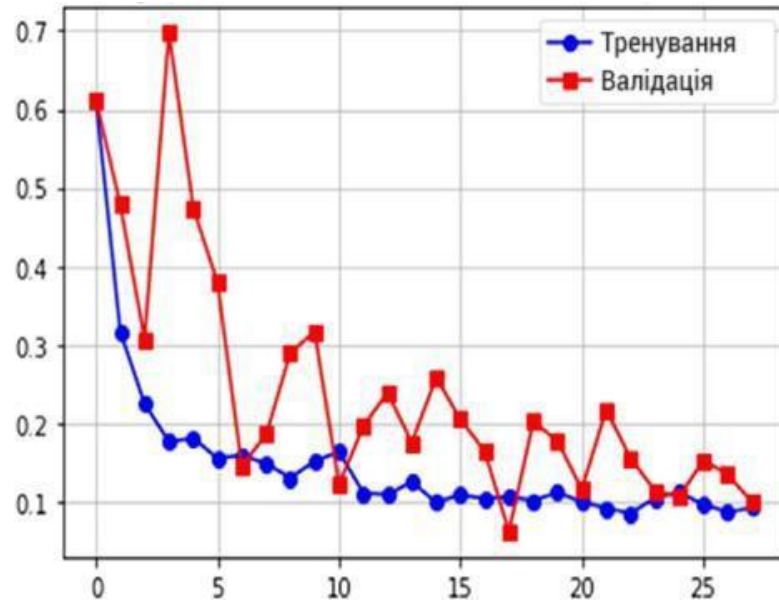


Рисунок 3.10 – Втрати моделі VGG- CapsNet для задачі класифікації на три класи

Втрата навчання нейронної мережі різко падає протягом перших п'яти ітерацій і стабілізується після очікування десяти ітерацій. Після очікування 10 епох навчання припиняється після 29 епох. Модель мережевої капсули візуальної геометрії VGG-CapsNet. Матриця плутанини для проблеми бінарної класифікації сформована на основі роботи [14], як показано на рисунку 3.11.

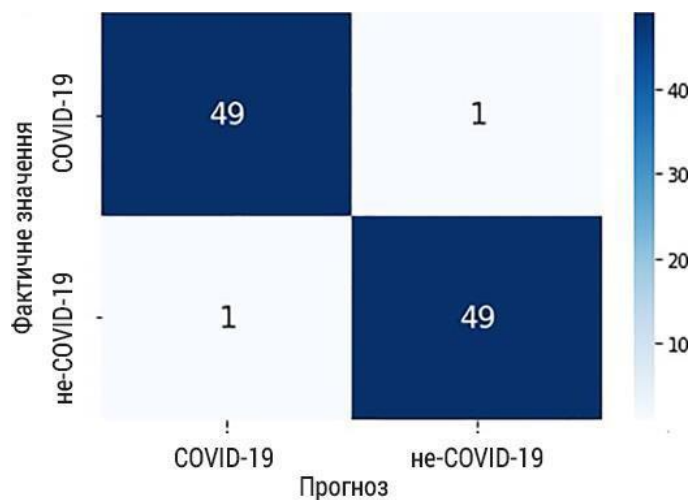


Рисунок 3.11 – Матриця сплутування моделі VGG-CapsNet для двокласової задачі

Матриця сплутування моделі капсульної мережі візуальної геометрії

VGG-CapsNet для трьохкласової задачі подана на рисунку 3.12



Рисунок 3.12 – Матриця сплутування моделі VGG-CapsNet для класифікації на три класи

У таблиці 3.2 показано показники ефективності моделей CNN-CapsNet і VGG-CapsNet на основі експериментальних результатів [14] щодо трьох завдань класифікації COVID19, пневмонії та вірусної пневмонії.

Таблиця 3.2 – Показники продуктивності результатів експерименту для моделей CNN-CapsNet і VGG-CapsNet для трикласових задач

Клас	CNN-CapsNet			VGG-CapsNet		
	Точність	Запам'ятовуваність	Оцінка F1	Точність	Запам'ятовуваність	Оцінка F1
1	2	3	4	5	6	7
«COVID та SARS»	0.96	0.86	0.91	0.96	0.98	0.97
Звичайна пневмонія	0.89	1.00	0.94	0.87	0.96	0.91
1	2	3	4	5	6	7
Вірусна пневмонія	0.88	0.86	0.87	0.92	0.82	0.87
Точність	0.91			0.92		

### 3.4 Аналіз результатів моделювання

Карту функцій капсульної моделі згорткової нейронної мережі CNNCapsNet [14] для задачі бінарної класифікації COVID та SARS показано на рисунку 3.13.

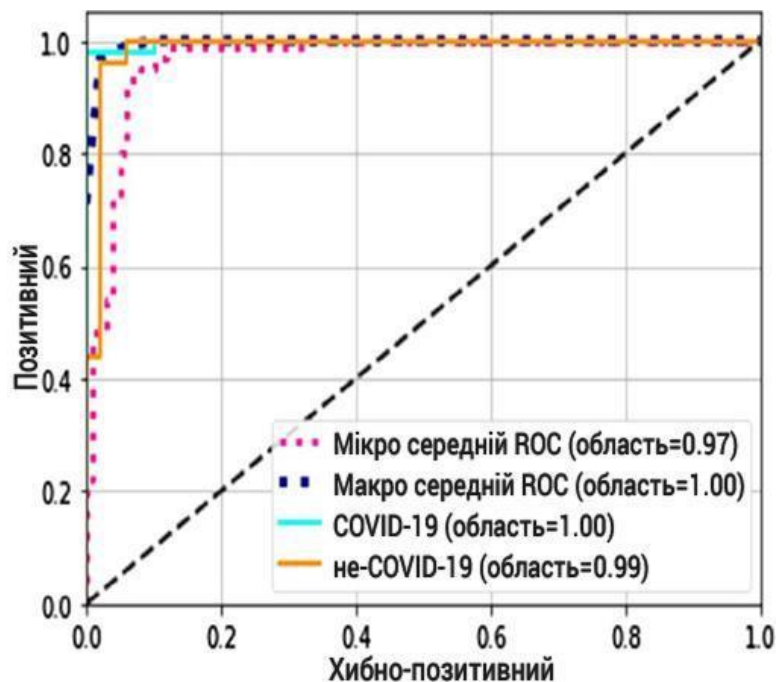


Рисунок 3.13 – Характеристики моделі CNNCapsNet для задачі бінарної класифікації

На рисунку 3.14 показано карту функцій групової капсульної мережевої моделі візуальної геометрії VGG-CapsNet [14] для завдання двійкової класифікації COVID та SARS.

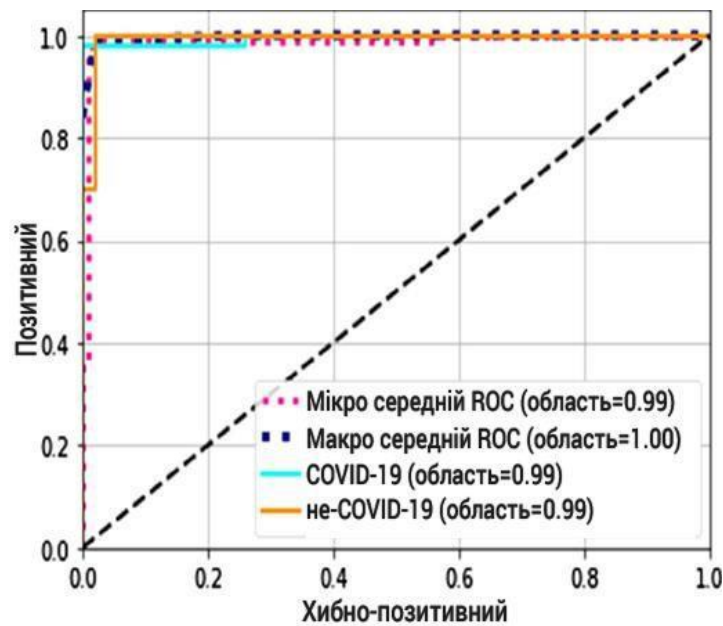


Рисунок 3.14 – Характеристики моделі VGG-CapsNet для задачі бінарної класифікації

Карта характеристик продуктивності капсульної моделі згорткової нейронної мережі CNNCapsNet, метрики AUC, мікро- та макросередні метрики продуктивності для завдання класифікації на три класи [14] показані на рисунку 3.15.

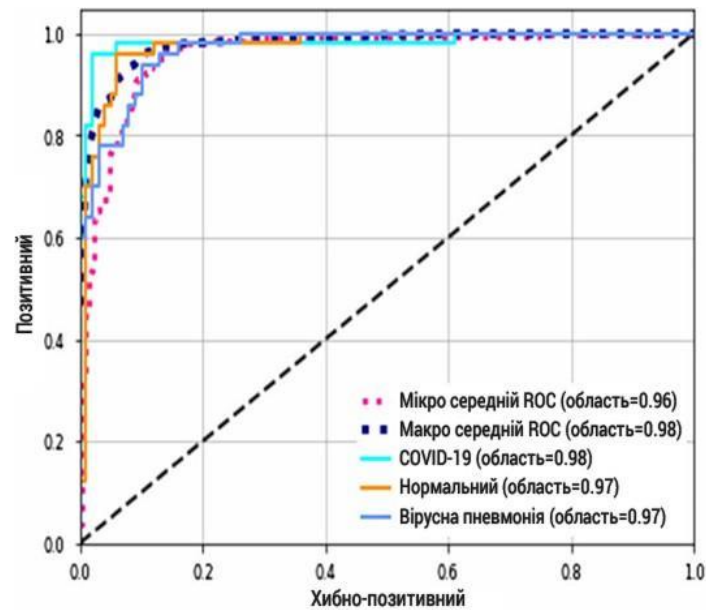


Рисунок 3.15 – Характеристики моделі CNNCapsNet для класифікації на три класи

На рисунку 3.16 показано візуальну геометричну карту характеристик продуктивності, метрики AUC, мікро-середнє та макросереднє метрики ефективності групи капсульних мережеских моделей VGG-CapsNet для класифікації завдань на три категорії [14].

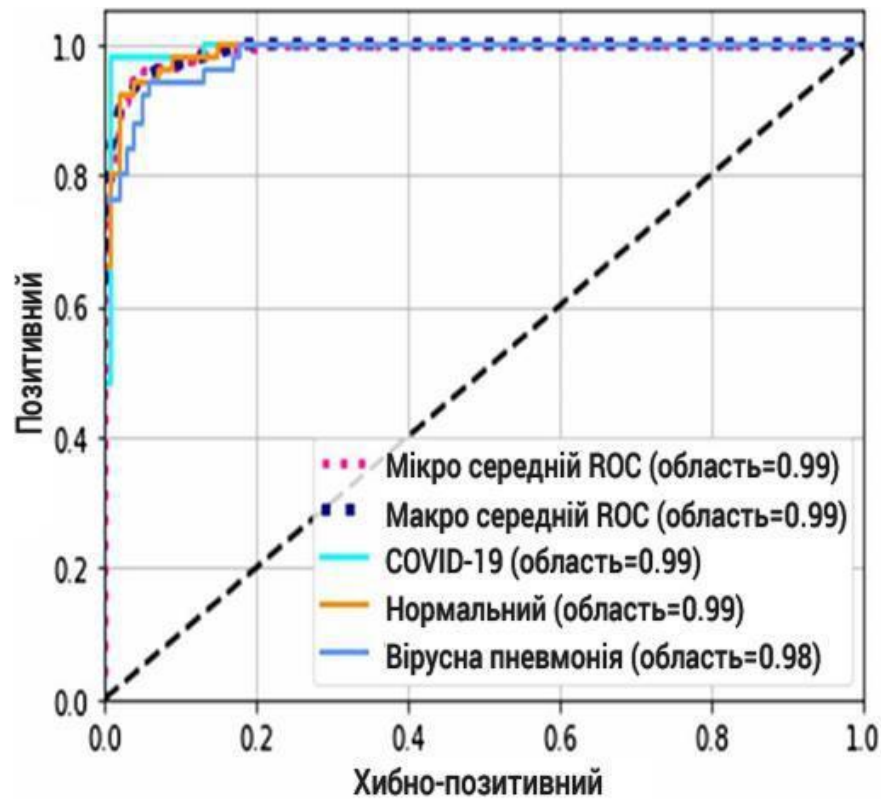


Рисунок 3.16 – Характеристики моделі VGG-CapsNet для класифікації на три класи

На додаток до ділянки ROC на рис. На рисунку 3.16 показано AUC, мікросереднє та макросереднє значення ефективності для категорій COVID та SARS, «нормальної» та «вірусної пневмонії» відповідно. Було проведено чотири незалежні експерименти з використанням моделей нейронної мережі CNN-CapsNet і VGG-CapsNet. Результати представлені в попередньому абзаці у вигляді матриць плутанини та ефективності. На основі аналізу результатів [14] можна зробити висновок, що модель групової капсульної мережі VGG-CapsNet для візуальної геометрії забезпечує вищу точність порівняно з моделлю згорткової нейронної мережі CNN-CapsNet. Перегляньте результати завдань бінарної класифікації, зокрема «COVID і SARS»/«Non-COVID19» і трикласову класифікацію «COVID і SARS»/«нормальна»/«вірусна пневмонія». Для задач бінарної класифікації модель капсульної мережі візуальної геометрії VGG-CapsNet дозволяє забезпечити точність 97%. Це вище, ніж 95% точність моделі CNN-CapsNet. Модель капсульної мережі візуальної геометрії VGG-CapsNet досягає 92% точності для завдань класифікації трьох класів, що вище, ніж 91%



точності моделі CNN-CapsNet. Графіки ROC, показані на рисунках 3.13-3.16, підтверджують, що модель VGG-CapsNet перевершує модель CNN-CapsNet. У моделі VGG-CapsNet для задачі бінарної класифікації мікросередня площа становить 0,99, а макросередня область — 1,0. Порівняно з моделлю ConvNet показник мікросередньої площі дещо покращений і становить 0,97. У задачі класифікації трьох класів моделі VGG-CapsNet мікро-усереднена область становить 0,99, а макро-усереднена область — 0,99. Ці показники дещо вдосконалені порівняно з моделями ConvNet. Для моделей згорткової нейронної мережі вони становлять 0,96 і 0,98. На основі порівняння графіків продуктивності та ROC можна зробити висновок, що модель VGG-CapsNet надійніша за модель CNN-CapsNet.

Раннє виявлення громадян, інфікованих COVID та SARS, має вирішальне значення для швидкого реагування та запобігання поширенню пандемії. У третій частині кваліфікаційної роботи наводяться описи обчислювальних експериментів та результати моделювання. Розглянута модель на основі глибокого навчання призначена для автоматичної ідентифікації пацієнтів, інфікованих COVID та SARS, виявлення здорових громадян і хворих на вірусну пневмонію. Дві моделі глибокого навчання CNN-CapsNet і

VGGCapsNet. Перша модель була розроблена на основі глибоких CNN, щоб мінімізувати властиві CNN недоліки. CNN-CapsNet можна використовувати в системах комп'ютерного зору. Він був розроблений шляхом заміни звичайних скалярних активацій векторами та різними стилями маршрутизації нейронів. Завдяки інтеграції CNN-CapsNet з моделями CNN і VGG16 покращено обчислювальну потужність початкової карти ознак. Проведіть аналіз продуктивності, щоб вибрати найкращу модель. Результати аналізу показують, що модель VGG-CapsNet перевершує модель CNN-CapsNet.

## 4. ЕКОНОМІЧНИЙ РОЗДІЛ

### 4.1 Технологічний аудит розробленої нейромережевої системи для класифікації медичних зображень

Як було підкреслено раніше, сьогодні світ перебуває у розпалі епідемії, викликаних поширенням різних штамів коронавірусної інфекції, COVID, SARS та інших, що створює серйозну небезпеку для життя і здоров'я населення України, особливо в умовах воєнного стану, викликаного безпідставною агресією росії проти України. Тому одним із головних напрямів державної політики України в сучасних умовах є не тільки захист населення від поширення інфекційних захворювань, а і збереження і зміцнення здоров'я всіх верств населення, а самі медичні установи мають стають об'єктами підвищеної уваги держави. Суттєву допомогу у розв'язанні цих питань є впровадження в роботі медичних закладів всіх рівнів автоматизованих систем управління медичними послугами.

Тому метою нашої магістерської кваліфікаційної роботи було проведення досліджень методів аналітичного опрацювання медичних зображень та підвищення рівня цілісності і ефективності нейромережевої системи, яка використовується в медичних установах для аналізування медичної графічної інформації і насамперед медичних зображень, особливо в умовах дії пандемії.

Для досягнення поставленої цілі нами було: проаналізовано стан досліджень у досліджуваній предметній області; вивчено існуючі на даний час практичні реалізації на основі згорткових нейронних мереж; проаналізовано моделі виявлення інфекційних захворювань з використанням згорткових нейронних мереж; досліджено обчислювальні експерименти на основі згорткових нейронних мереж для задач опрацювання медичних зображень в умовах пандемії тощо.

Для встановлення комерційного потенціалу розробленої нами нейромережевої системи для класифікації медичних зображень було запрошено 3-х експертів – фахівців у цій галузі знань: кандидатів технічних наук професорів Бикова М.М., Ковтуна В.В. та к.т.н. доцента Ковалюка О.О.

Встановлення потенційних можливостей комерційного використання нашої розробки було здійснено за критеріями, наведеними в таблиці 4.1,

Таблиця 4.1 – Рекомендовані критерії оцінювання технічного рівня та комерційного потенціалу будь-якої розробки і їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

	продукту	коштів та часу	часу		
--	----------	----------------	------	--	--

Запрошені експерти оцінили розроблену нами нейромережеву систему для класифікації медичних зображень таким чином (див. табл. 5.2):

Таблиця 4.2 – Результати технологічного аудиту розробленої нейромережевої системи (за шкалою оцінювання 0-1-2-3-4)

Критерії	Прізвище, ініціали експертів		
	Биков М.М.	Ковтун В.В.	Ковалюк О.О.
	Бали, що їх виставили експерти:		
1	3	4	3
2	3	3	3
3	3	3	4
4	3	4	4
5	3	3	3
6	3	3	4
7	3	3	4
8	4	3	3
9	4	4	3
10	3	3	4
11	3	4	4
12	3	3	3
Сума балів	СБ <sub>1</sub> = 38	СБ <sub>2</sub> = 40	СБ <sub>3</sub> = 42
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{38 + 40 + 42}{3} = \frac{120}{3} = 40$		

Встановлення комерційного потенціалу розробленої нейромережевої системи будемо здійснювати на основі рекомендацій, наведених в таблиці 5.3 [1].

Таблиця 5.3 – Рівні комерційного потенціалу будь-якої наукової розробки

Середньоарифметична сума балів $\overline{СБ}$ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 40 балів, то це свідчить, що розроблена нами нейромережева система

для класифікації медичних зображень має рівень комерційного потенціалу, який вважається «вище середнього».

Це пояснюється тим, що розроблена нами нейромережева система для класифікації медичних зображень передбачає подальший розвиток моделі згорткової капсульної нейронної мережі та моделі візуальної геометричної капсульної мережі і може бути використана при проектуванні і створенні прототипів моделей нейронної мережі Convolution Capsule і Visual Geometric Capsule Network.

#### 4.2 Розрахунок витрат на розробку нейромережевої системи для класифікації медичних зображень

При дослідженні та розробці нової нейромережевої системи для класифікації медичних зображень були зроблені такі витрати.

А). Основна заробітна плата  $Z_o$  розробників, яка визначається за формулою:

$$Z_o = \frac{M}{T_p} \cdot t \quad \text{грн,}$$

(4.1)

де  $M$  – місячний посадовий оклад розробника, грн; прийmemo, що

$M = (6700 \dots 20000)$  грн/місяць;

$T_p$  – число робочих днів в місяці; прийmemo  $T_p = 24$  дні;

$t$  – число днів роботи розробників.

Зроблені розрахунки зведемо до таблиці 5.4:

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник магістерської роботи	20000	833,83	20 годин	$\approx 2778$
2. Магістрант-студент-виконавець	2000 (беремо 6700)	279,17	85	$\approx 23729$

3. Консультант з економічної частини	17500	729,17	1,5 години	≈ 183 (при 6-годинному робочому дні)
Загалом				$Z_0 = 26690$ грн

Б). Додаткова заробітна плата  $Z_d$  розробників розраховується як (10...12)% від величини їх основної заробітної плати, тобто:

$$Z_d = \alpha \cdot Z_0 = (0,1...0,12) \cdot Z_0. \quad (4.2)$$

Прийmemo, що  $\alpha = 0,1$ . Тоді для нашого випадку отримаємо:

$$Z_d = 0,1 \times 26690 = 2669 \text{ грн.}$$

В). Нарахування на заробітну плату НЗП<sub>зп</sub> розробників (дослідників) розраховуються за формулою:

$$\text{НЗП}_{\text{зп}} = (Z_0 + Z_d) \cdot \frac{\beta}{100}, \quad (4.3)$$

де  $\beta$  – ставка обов'язкового єдиного внеску на державне соціальне страхування, %.  $\beta = 22\%$ . Тоді:

$$\text{НЗН}_{\text{зп}} = (26690 + 2669) \times 0,22 = 6458,98 \approx 6459 \text{ грн.}$$

Г). Амортизація основних засобів  $A$ , які використовувались під час виконання даної роботи:

$$A = \frac{Ц \cdot N_a}{100} \cdot \frac{T}{12} \quad \text{грн,} \quad (4.4)$$

де  $Ц$  – загальна балансова вартість основних засобів, грн;

$N_a$  – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що  $N_a = (5...25)\%$ ;

$T$  – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 5.5.

Таблиця 4.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютерна техніка, обладнання тощо	58000	25	3 (при 50% використанні)	1812 ≈ 1813
2. Приміщення університету, кафедри	18500	4	3 при 20% використанні	37
Всього				- A = 1850 грн

Д). Витрати на матеріали  $M$  розраховуються за формулою:

$$M = \sum_1^n H_i \cdot \Pi_i \cdot K_i - \sum_1^n V_i \cdot \Pi_v \quad \text{грн.},$$

(4.5)

де  $H_i$  – витрати матеріалу  $i$ -го найменування, кг;  $\Pi_i$  – вартість матеріалу  $i$ -го найменування;  $K_i$  – коефіцієнт транспортних витрат,  $K_i = (1,1 \dots 1,15)$ ;  $V_i$  – маса відходів матеріалу  $i$ -го найменування;  $\Pi_v$  – ціна відходів матеріалу  $i$ -го найменування;  $n$  – кількість видів матеріалів.

Е). Витрати на комплектуючі  $K$  розраховуються за формулою:

$$K = \sum_1^n H_i \cdot \Pi_i \cdot K_i \quad \text{грн.},$$

(4.6)

де  $H_i$  – кількість комплектуючих  $i$ -го виду, шт.;  $\Pi_i$  – ціна комплектуючих  $i$ -го виду;  $K_i$  – коефіцієнт транспортних витрат,  $K_i = (1,1 \dots 1,15)$ ;  $n$  – кількість видів комплектуючих.

Під час виконання роботи загальні витрати на матеріали та комплектуючі склали приблизно 1800 грн.

Ж). Витрати на силову електроенергію  $V_e$  розраховуються за формулою:

$$V_e = \frac{B \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d},$$

(4.7)

де  $B$  – вартість 1 кВт-год. електроенергії, в 2022 р.  $B \approx 3,0$  грн/кВт;



$P$  – установлена потужність обладнання, кВт;  $P = 0,85$  кВт;

$\Phi$  – фактична кількість годин роботи обладнання, годин.

Прийmemo, що  $\Phi = 210$  годин;

$K_{\text{п}}$  – коефіцієнт використання потужності;  $K_{\text{п}} < 1 = 0,75$ .

$K_{\text{д}}$  – коефіцієнт корисної дії,  $K_{\text{д}} = 0,6$ .

Тоді витрати на силову електроенергію будуть дорівнювати:

$$V_e = \frac{V \cdot P \cdot \Phi \cdot K_{\text{п}}}{K_{\text{д}}} = \frac{3 \cdot 0,85 \cdot 210 \cdot 0,75}{0,6} = 669,37 \approx 670 \text{ грн.}$$

И). Інші витрати  $V_{\text{інш}}$  можна прийняти як (50...300)% від основної заробітної плати розробників, тобто:

$$V_{\text{інш}} = (0,5 \dots 3) \times Z_o. \quad (4.8)$$

Для нашого випадку отримаємо:

$$V_{\text{інш}} = 1,5 \times 26690 = 40035 \text{ грн.}$$

К). Сума всіх попередніх статей витрат складає витрати на виконання нашої роботи (безпосередньо розробником-магістрантом) –  $V$ .

$$V = 26690 + 2669 + 6459 + 1850 + 1800 + 670 + 40035 = 80173 \text{ грн.}$$

Л). Загальні витрати на розробку та можливе впровадження розробленої нами нейромережевої системи для класифікації медичних зображень  $V_{\text{заг}}$  розраховуються за формулою:

$$V_{\text{заг}} = \frac{V}{\beta},$$

(4.9)

де  $\beta$  – коефіцієнт, який характеризує етап (стадію) виконання цієї роботи.

Можна прийняти, що,  $\beta \approx 0,6$  [1], оскільки робота потребує подальшого доопрацювання.

$$\text{Тоді: } V_{\text{заг}} = \frac{80173}{0,6} = 133621,67 \text{ грн або приблизно 134 тисячі грн.}$$

Тобто прогнозовані загальні витрати на розробку та можливе впровадження розробленої нами нейромережевої системи для класифікації медичних зображень становлять приблизно 134 тисяч грн.

### **4.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки**

Економічний ефект від впровадження та можливої комерціалізації розробленої нами нейромережевої системи для класифікації медичних зображень пояснюється її значно кращими функціональними можливостями. Тому нашу розробку можна реалізовувати на ринку дещо дорожче, ніж аналогічні за функціями розробки. Так, якщо подібні за функціями розробки у 2022 році коштували на ринку приблизно \$9,5 тис (або приблизно 380 тис грн за курсом 1\$=40 грн), то нашу розробку можна буде реалізовувати приблизно за \$10 тисяч (400 тисяч грн) або на 20 тисяч грн дорожче.

Аналіз місткості ринку показує, що на сьогодні в Україні кількість реальних користувачів подібних систем може складати приблизно 20 осіб щороку. Це, насамперед, різні медичні заклади, науково-дослідні організації тощо. Тому можна очікувати зростання попиту на нашу розробку принаймні протягом 3-х років після її впровадження (після чого мають з'явитися нові, більш ефективні системи, розроблені насамперед фахівцями ВНТУ).

Тобто, якщо наша розробка буде впроваджена з 1 січня 2024 року (оскільки потребує що доопрацювання), то її результати будуть виявлятися протягом 2024-го, 2025-го та 2026-го років.

Прогноз зростання попиту на нашу розробку складає по роках:

- а) 2024 р. – приблизно +5 шт. до базового року;
- б) 2025 р. – +15 шт. до базового року;
- в) 2026 р. – +10 шт. до базового року.

Можливе збільшення чистого прибутку  $\Delta\Pi_i$ , що його може отримати потенційний інвестор від комерціалізації нашої розробки становитиме:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right),$$

(4.10)

де  $\Delta\Pi_o$  – покращення основного якісного показника від впровадження результатів розробки у цьому році. Для нашого випадку це є збільшення ціни реалізації нашої розробки  $\Delta\Pi_o = 400 - 380 = + 20$  тисяч грн;

$N$  – основний кількісний показник, який визначає обсяг діяльності у році до впровадження результатів розробки;  $N = 20$  шт.;

$\Delta N$  – покращення основного кількісного показника від впровадження результатів розробки. Таке покращення становитиме по роках, відповідно: у 2024 році – + 5 шт., у 2025 році +15 шт. та у 2026 році +10 шт. (до базового 2022 року);

$\Pi_o$  – основний якісний показник (тобто ціна), який визначає обсяг діяльності у році після впровадження результатів розробки, грн;  $\Pi_o = 400$  тисячі грн;

$n$  – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; для нашого випадку  $n = 3$ ;

$\lambda$  – коефіцієнт, який враховує сплату податку на додану вартість;  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати  $\rho = (0,2...0,5)$ ; візьмемо  $\rho = 0,5$ ;

$v$  – ставка податку на прибуток. У 2022-23 роках  $v = 18\%$ . У 2024 році також очікуємо на 18%.

Тоді можливе зростання чистого прибутку  $\Delta\Pi_1$  для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2024 р.) складе:

$$\Delta\Pi_1 = [20 \cdot 20 + 400 \cdot 5] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 820 \text{ тис. грн.}$$

Можливе зростання чистого прибутку  $\Delta\Pi_2$  для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2025 р.) року складе:

$$\Delta\Pi_2 = [20 \cdot 20 + 400 \cdot 15] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 2187 \text{ тис. грн.}$$

Можливе зростання чистого прибутку  $\Delta\Pi_3$  для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2026 р.) року складе:

$$\Delta\Pi_3 = [20 \cdot 20 + 400 \cdot 10] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 1503 \text{ тис. грн.}$$

Приведена вартість зростання всіх чистих прибутків від можливого впровадження нашої розробки становитиме:

$$\text{ПП} = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.11)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

$t$  – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку  $t = 3$  роки;

$\tau$  – ставка дисконтування. Прийнемо  $\tau = 0,10$  (10%);

$t$  – період часу від моменту початку розроблення нейромережевої системи для класифікації медичних зображень до моменту отримання можливих чистих прибутків.

Тоді приведена вартість зростання всіх можливих чистих прибутків ПП, що їх може отримати потенційний інвестор від комерціалізації нашої розробки, складе:

$$\text{ПП} = \frac{820}{(1 + 0,1)^2} + \frac{2187}{(1 + 0,1)^3} + \frac{1503}{(1 + 0,1)^4} \approx 678 + 1644 + 1027 = 3349 \text{ тисяч грн.}$$

Теперішня вартість інвестицій PV, що повинні бути вкладені для реалізації нашої розробки:  $PV = (1,0 \dots 5) \times V_{\text{заг}}$ .

Для нашого випадку  $PV = (1,0 \dots 5) \times 134 = 5,0 \times 134 = 670$  тисяч грн.

Розраховуємо абсолютний ефект від можливих вкладених інвестицій  $E_{\text{абс}}$ .

$$E_{\text{абс}} = \text{ПП} - PV, \quad (4.12)$$

де ПП – приведена вартість збільшення всіх чистих прибутків для інвестора від можливого впровадження нашої розробки, грн;

PV – теперішня вартість інвестицій PV = 98 тисяч грн.

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$$E_{\text{абс}} = 3349 - 670 = 2679 \text{ тисяч грн.}$$

Оскільки  $E_{\text{абс}} > 0$ , то комерціалізація нашої розробки може бути доцільною.

Далі розрахуємо внутрішню дохідність  $E_{\text{в}}$  вкладених інвестицій:

$$E_{\text{в}} = T_{\text{ж}} \sqrt[4]{1 + \frac{E_{\text{абс}}}{PV}} - 1,$$

(4.13)

де  $E_{\text{абс}}$  – абсолютний ефект вкладених інвестицій;  $E_{\text{абс}} = 2679$  тис. грн;

PV – теперішня вартість початкових інвестицій PV = 670 тис. грн;

$T_{\text{ж}}$  – життєвий цикл розробки, роки.

$T_{\text{ж}} = 4$  років (2023-й, 2024-й, 2025-й, 2026-й роки)

Для нашого випадку отримаємо:

$$E_{\text{в}} = \sqrt[4]{1 + \frac{2679}{670}} - 1 = \sqrt[4]{1 + 3,9985} - 1 = \sqrt[4]{4,9985} - 1 = 1,495 - 1 = 0,495 = 49,5\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційному інвестору не вигідно буде займатися комерціалізацією нашої розробки.

Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування  $\tau_{\text{мін}}$  визначається за формулою:

$$\tau_{\text{мін}} = d + f,$$

(4.14)

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні  $d = (0,10 \dots 0,12)$ ;

$f$  – показник, що характеризує ризикованість вкладень;  $f = (0,05 \dots 0,30)$ .

Для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,12 + 0,30 = 0,42 \text{ або } \tau_{\text{мін}} = 42\%.$$

Оскільки величина  $E_{\text{в}} = 49,5\% > \tau_{\text{мін}} = 42\%$ , то потенційний інвестор у принципі може бути зацікавлений у фінансуванні та комерціалізації

розробленої нами нейромережевої системи для класифікації медичних зображень.

Далі розраховуємо термін окупності коштів, вкладених у можливу комерціалізацію розробленої нами нейромережевої системи для класифікації медичних зображень.

Термін окупності  $T_{ок}$  розраховується за формулою:

$$T_{ок} = \frac{1}{E_B}.$$

(4.15)

Для нашого випадку термін окупності  $T_{ок}$  коштів становитиме:

$$T_{ок} = \frac{1}{0,495} = 2,02 \text{ років} < 3 \text{ років},$$

що свідчить про потенційну доцільність комерціалізації розробленої нами нейромережевої системи для класифікації медичних зображень.

Далі проведено моделювання залежності величини внутрішньої дохідності вкладених потенційних інвестицій від рівня інфляції в країні. Як відомо, на наступні роки, на жаль, прогнозується високий рівень інфляції (в межах 30% і вище), що обумовлюється військовою агресією росії проти України.

Прийнявши рівень інфляції у 30% отримаємо:

$$ПП = \frac{820}{(1+0,3)^2} + \frac{2187}{(1+0,3)^3} + \frac{1503}{(1+0,3)^4} \approx 485 + 995 + 526 = 2006 \text{ тисяч грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$E_{абс} = 2006 - 670 = 1366 \text{ тисяч грн.}$$

Внутрішня дохідність  $E_B$  вкладених інвестицій становитиме:

$$E_B = T_{ж} \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1,$$

де  $E_{абс}$  – абсолютний ефект вкладених інвестицій;  $E_{абс} = 1366$  тисяч грн;

$PV$  –теперішня вартість початкових інвестицій  $PV = 670$  тисяч грн.

Для нашого випадку отримаємо:

$$E_B = \sqrt[4]{1 + \frac{1366}{670}} - 1 = \sqrt[4]{1 + 2,0388} - 1 = \sqrt[4]{3,0388} - 1 = 1,32 - 1 = 0,323 = 32,0\%.$$

Зроблені розрахунки у вигляді графіків наведено на рис. 4.1.

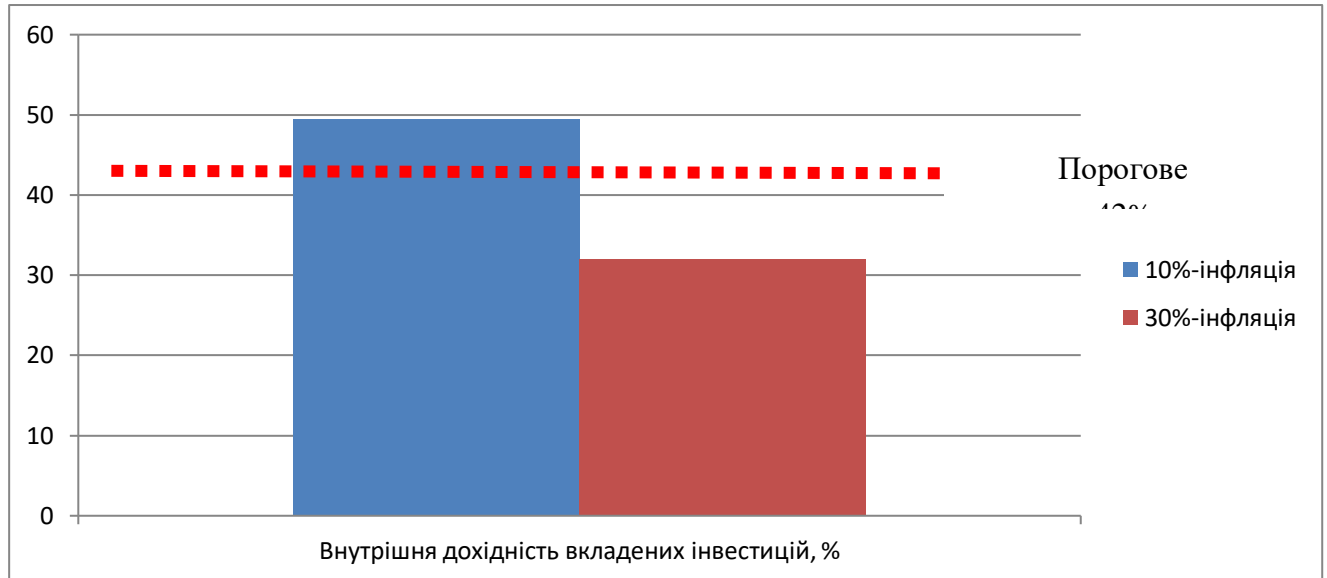


Рисунок 4.1 – Моделювання залежності величини внутрішньої дохідності потенційних інвестицій від рівня інфляції в країні

Аналіз графіка на рис 5.1 показує, що при рівні інфляції в 10% величина внутрішньої дохідності інвестицій становить  $E_B = 49,5\%$ , що більше порогового значення  $\tau_{\min} = 42\%$  і тому комерціалізація нашої розробки може бути доцільною; при рівні інфляції в 30% величина внутрішньої дохідності можливих інвестицій становить  $E_B = 32\%$ , що менше порогового значення  $\tau_{\min} = 42\%$  і комерціалізація нашої розробки (при нинішніх умовах) буде для інвестора проблематичною. Для прийняття остаточного рішення потрібні будуть додаткові обґрунтування і рішення (наприклад, підняти ціну на реалізацію розробленої нами нейромережевої системи для класифікації медичних зображень. тощо).

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю:

Показники	Задані у ТЗ	Досягнуті у магістерській	Висновок

		кваліфікаційній роботі	
1. Витрати на розробку	Не більше 150 тис. грн	134 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження розробки (в майбутній вартості грошей), тисяч грн	Не менше 2500 тисяч грн (за три роки)	2679 тисяч грн	Виконано
3. Внутрішня дохідність інвестицій, %	не менше 42%	49,5%	Досягнуто
4. Термін окупності інвестицій, роки	до 3-ти років	2,02 років	Виконано

Таким чином, основні техніко-економічні показники розробленої нами нейромережевої системи для класифікації медичних зображень, визначені у технічному завданні, виконані.

#### ЛІТЕРАТУРА

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



## ВИСНОВКИ

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр»:

- Описано процес розвиток епідеміологічної загрози.
- Висвітлено стан досліджень в галузі опрацювання медичних зображень.
- Проведено наукові розвідки про використання CNN для опрацювання медичних зображень в умовах пандемії.

В другому розділі кваліфікаційної роботи:

- Описано раннє виявлення COVID та SARS з використанням CNN.
- Розглянуто згорткові нейронні мережі.
- Досліджено гібридну модель CNN.
- Висвітлено капсульні мережі.
- Досліджено нейромережну архітектуру VGG-CapsNet.
- Подано опис набору даних зображень.

В третьому розділі кваліфікаційної роботи:

- Описано обчислювальний експеримент та подано результати моделювання.
- Досліджено модель CNN-CapsNet для виявлення COVID та SARS.
- Наведено результати використання моделі VGG-CapsNet для виявлення COVID та SARS.
- Проаналізовано результати моделювання.

## ПЕРЕЛІК ДЖЕРЕЛ

- 1 Guo Y.R., Cao Q.D., Hong Z.S., et al. The origin, transmission and clinical therapies on corona virus disease 2019 (COVID та SARS) outbreak—an update on the status. *Mil Med Res.* 2020;7(1):1-10.
- 2 Worldometer. COVID та SARS CORONAVIRUS PANDEMIC. Coronavirus Cases. <https://www.worldometers.info/coronavirus/>.
- 3 Woolhouse M., Scott F., Hudson Z., Howey R., Chase-Topping M. Human viruses: discovery and emergence. *Philos Trans R Soc B Biol Sci.* 2012;367(1604):2864-2871.
- 4 Cascella M., Rajnik M., Cuomo A., Dulebohn S.C., di Napoli R. Features, evaluation and treatment corona virus (COVID та SARS). Statpearls [internet]. Treasure Island(FL): StatPearls Publishing; 2020. <https://www.ncbi.nlm.nih.gov/books/NBK554776/>.
- 5 Duda, O., et al, Selection of Effective Methods of Big Data Analytical Processing in Information Systems of Smart Cities. *CEUR Workshop Proceedings* 2631, pp. 68-78. 2020.
- 6 Bodnarchuk I., Duda O., Kharchenko A., Kunanets N., Matsiuk O., Pasichnyk V. Choice method of analytical information-technology platform for projects associated to the smart city class. *ICTERI 2020 ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I: Main Conference p.317-330.*
- 7 Singhal T. A review of corona virus disease-2019 (COVID та SARS). *Indian J Pediatr.* 2020;87(4):281-286.
- 8 Kong W., Agarwal P.P. Chest imaging appearance of COVID та SARS infection. *Radiol Cardiothorac Imaging.* 2020;2(1):e200028.
- 9 R. Baig, M. Bibi, A. Hamid, S. Kausar, S. Khalid, Deep learning approaches towards skin lesion segmentation and classification from dermoscopic images-a review, *Current Med. Imag.* 16 (2020) 513–533.

- 10 Duda, O., Pasichnyk, V., Kunanets, N., Antonii, R., Matsiuk, O. Multidimensional Representation of COVID та SARS Data Using OLAP Information Technology. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2020, 2, pp. 277–280, 9321889.
- 11 LeCun Y., Bengio Y., Hinton G. Deep learning. Nature. 2015;521 (7553):436-444.
- 12 Jain A, Tiwari S, Sapra V. Hands on Deep Learning with Python Programming. Germany: Lambert Academic Publishing House; 2018 978-613-9-91551-4. <https://www.lap-publishing.com/catalog/details/store/gb/book/978-613-9-91551-4/hands-on-deeplearning-with-python-programming>.
- 13 Iesmantas T., Alzbutas R. Convolutional capsule network for classification of breast cancer histology images. International Conference Image Analysis and Recognition. Cham, Switzerland: Springer; 2018:853-860.
- 14 Tiwari S., Shamik, Anurag J.. "Convolutional capsule network for COVID-19 detection using radiography images." International Journal of Imaging Systems and Technology 31.2 (2021): 525-539.
- 15 Apostolopoulos I.D., Mpesiana TA. COVID та SARS: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med. 2020;43(2): 635–640.
- 16 Hall L.O., Paul R., Goldgof D.B., Goldgof G.M. Finding COVID та SARS from chest X-Rays using deep learning on a small dataset. arXiv preprint arXiv:2004.02060; 2020.
- 17 Zhang J., Xie Y., Li Y., Shen C., Xia Y.. COVID та SARS screening on chest X-Ray images using deep learning based anomaly detection. arXiv preprint arXiv:2003.12338; 2020.
- 18 Ucar F., Korkmaz D. COVIDiagnosis-net: deep BayesSqueezeNet based diagnostic of the corona virus disease 2019 (COVID та SARS) from X-ray images. Med Hypotheses. 2020;140 (2020):1–12.
- 19 Khalifa N.E.M., Taha M.H.N., Hassanien A.E., Elghamrawy S.. Detection of corona virus (COVID та SARS) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using

chest X- Ray dataset. arXiv preprint arXiv:2004.01184; 2020.

20 Salman F.M., Abu-Naser SS, Alajrami E, Abu-Nasser BS, Alashqar BA, COVID ta SARS Detection Using Artificial Intelligence; United States: The DSpace Institutional Digital Repository System; 2020. <http://dstore.alazhar.edu.ps/xmlui/handle/123456789/587>.

21 Sethy P.K., Behera S.K. Detection of corona virus disease (COVID19) based on deep features. Preprints, 2020030300; 2020.

22 Maghdid H.S, Asaad A.T., Ghafoor K.Z., Sadiq A.S., Khan M.K. Diagnosing COVID ta SARS pneumonia from X-Ray and CT images using deep learning and transfer learning algorithms. arXiv preprint arXiv:2004.00038; 2020.

23 Bassi P.R., Attux R. A deep convolutional neural network for COVID-19 detection using chest X-rays. arXiv preprint arXiv: 2005.01578; 2020.

24 Ozturk T., Talo M., Yildirim E.A., Baloglu U.B., Yildirim O., Acharya U.R. Automated detection of COVID ta SARS cases using deep neural networks with X- ray images. *Comput Biol Med.* 2020;121:103792.

25 Mei X., Lee H.C., Diao K.Y., et al. Artificial intelligence-enabled rapid diagnosis of patients with COVID ta SARS. *Nat Med.* 2020;26 (8):1224-1228.

26 Li L., Qin L., Xu Z., et al. Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: evaluation of the diagnostic accuracy. *Radiology.* 2020;296(2):E65.

27 Toraman S., Alakus T.B., Türkoglu I. Convolutional CapsNet: a novel artificial neural network approach to detect COVID ta SARS disease from X-ray images using capsule networks. *Chaos Solitons Fractals.* 2020;140:110122.

28 Shoeibi A., Khodatars M., Alizadehsani R., et al. Automated detection and forecasting of COVID ta SARS using deep learning techniques: a review. arXiv preprint arXiv:2007.10785; 2020.

29 I.U. Khan, N. Aslam, A deep-learning-based framework for automated diagnosis of COVID ta SARS using X-ray images, *Information* 11 (2020) 419.

30 A.S.B. Aman Jaiswal, Analysis of deep learning algorithms on COVID-19 radiography database, *Int. J. Adv. Sci. Technol.* 29 (2020) 1268–1275.

31 M. Nour, Z. Cömert, K. Polat, A novel medical diagnosis model for

COVID19 infection detection based on deep features and Bayesian optimization, *Appl. Soft Comput.* (2020) 106580.

32 M.E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, Can AI help in screening viral and COVID ta SARS pneumonia?, 2020, ArXiv preprint arXiv: 2003.13145.

33 Aslan M.F., et al. "CNN-based transfer learning–BiLSTM network: A novel approach for COVID ta SARS infection detection." *Applied Soft Computing* 98 (2021): 106912.

34 S. Asif, Y. Wenhui, Automatic detection of COVID ta SARS using X-ray images with deep convolutional neural networks and machine learning, *medRxiv* (2020).

35 M. Toğaçar, B. Ergen, Z. Cömert, COVID ta SARS detection using deep learning models to exploit social mimic optimization and structured chest X-ray images using fuzzy color and stacking approaches, *Comput. Biol. Med.* (2020) 103805.

36 F. Ucar, D. Korkmaz, COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnostic of the coronavirus disease 2019 (COVID ta SARS) from X-ray images, *Med. Hypotheses* (2020) 109761.

37 T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID ta SARS cases using deep neural networks with X-ray images, *Comput. Biol. Med.* (2020) 103792.

38 A.I. Khan, J.L. Shah, M.M. Bhat, Coronet: A deep neural network for detection and diagnosis of COVID ta SARS from chest x-ray images, *Comput. Methods Programs Biomed.* (2020) 105581.

39 A. Sharma, S. Rani, D. Gupta, Artificial intelligence-based classification of chest X-ray images into COVID ta SARS and other infectious diseases, *Int. J. Biomed. Imaging* 2020 (2020).

40 A. Narin, C. Kaya, Z. Pamuk, Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks, 2020, ArXiv preprint arXiv:2003.10849.

- 41 M. Singh, S. Bansal, S. Ahuja, R.K. Dubey, B.K. Panigrahi, N. Dey, Transfer Learning Based Ensemble Support Vector Machine Model for Automated COVID ta SARS Detection using Lung Computerized Tomography Scan Data, 2020.
- 42 G. Liu, J. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification, *Neurocomputing* 337 (2019) 325–338.
- 43 L. Wang, A. Wong, COVID-Net: A tailored deep convolutional neural network design for detection of COVID ta SARS cases from chest X-ray images, 2020, ArXiv preprint arXiv:2003.09871.
- 44 A. Gupta, Anjum, S. Gupta, R. Katarya, InstaCovNet-19: A Deep learning classification model for the detection of COVID ta SARS patients using chest X-ray, *Appl. Soft Comput.* (2020) 106859.
- 45 X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, L. Yu, Q. Ni, Y. Chen, J. Su, A deep learning system to screen novel coronavirus disease 2019 pneumonia, *Engineering* (2020).
- 46 P.K. Sethy, S.K. Behera, Detection of coronavirus disease (covid-19) based on deep features, *Preprints 2020030300* (2020) 2020.
- 47 E.E.-D. Hemdan, M.A. Shouman, M.E. Karar, Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in X-ray images, 2020, ArXiv preprint arXiv:2003.11055.
- 48 M. Rahimzadeh, A. Attar, A new modified deep convolutional neural network for detecting COVID ta SARS from X-ray images, 2020, ArXiv preprint arXiv:2004.08052.
- 49 P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K.N. Plataniotis, A. Mohammadi, Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images, 2020, ArXiv preprint arXiv:2004.02696.
- 50 I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from xray images utilizing transfer learning with convolutional neural networks, *Phys. Eng. Sci. Med.* (2020) 1.
- 51 M. Farooq, A. Hafeez, Covid-resnet: A deep learning framework for screening of covid19 from radiographs, 2020, ArXiv preprint arXiv:2003. 14395.
- 52 A.M. Dawud, K. Yurtkan, H. Oztoprak, Application of deep learning in

neuroradiology: Brain haemorrhage classification using transfer learning, *Comput. Intell. Neurosci.* 2019 (2019).

53 D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, ArXiv preprint arXiv:1412.6980.

54 Tiwari S. A blur classification approach using deep convolution neural network. *Int J Inf Syst Model Des.* 2020;11(1):93-111.

55 Hinton G.E., Krizhevsky A., Wang S.D. Transforming autoencoders. *International Conference on Artificial Neural Networks.* Berlin, Heidelberg: Springer; 2011:44-51.

56 Sabour S., Frosst N., Hinton G.E. Dynamic routing between capsules. *Adv Neural Inf Proces Syst.* 2017;10(3):3856-3866.

57 Hinton G.E., Sabour S., Frosst N. Matrix Capsules with EM Routing; 2018:1–15. <https://openreview.net/forum?id=HJWLFgWRb&noteId=rk5MadsMf&noteId=rk5MadsMf>.

58 Tiwari S. Dermatoscopy using multi-layer perceptron, convolution neural network, and capsule network to differentiate malignant melanoma from benign nevus. *Int J Healthc Inf Syst Inform.* 2021;16(3):58-73.

59 Dadario A.M.V. COVID та SARS X rays. Kaggle. <https://doi.org/10.34740/KAGGLE/DSV/1019469>.

60 Juba B., Le H.S. Precision-recall versus accuracy and the role of large data sets. *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol 33; 2019:4039-4048.

61 Дії роботодавця, якщо у працівника виявлено коронавірус. <https://oppb.com.ua/news/diyi-robotodavcsya-yakshcho-u-pracivnyka-vuyavleno-koronavirus>.

62 Постанови Кабінету міністрів України від 22.07.2020 р. № 641. <https://zakon.rada.gov.ua/laws/show/641-2020-%D0%BF#Text>.

63 Законом України Про загальнообов'язкове державне соціальне страхування від 23.09.1999 р. № 1105-XIV. <https://zakon.rada.gov.ua/laws/show/1105-14#Text>.

64 Левченко, Олег Григорович, et al. "Безпека життєдіяльності та

цивільний захист." (2019).

65 Січко, Віктор Михайлович, and Анатолій Миколайович Тубальцев.  
"Безпека життєдіяльності." (2019).



**ДОДАТКИ**

## Додаток А (обов'язковий)

## Технічне завдання

ЗАТВЕРДЖЕНО

В.о. зав. кафедри АІТ

\_\_\_\_\_ Бісікало О. В.

«\_\_» \_\_\_\_\_ 2022 р

## ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу «Розробка глибокої  
нейромережевої системи для класифікації медичних зображень»

08-02.МКР.001.00.000 ТЗ

Керівник роботи:

к.т.н., доц. АІТ

Кулик Я.А.

«\_\_» \_\_\_\_\_ 2022 р.

Виконавець:

ст. гр. 1АКІТ-21м Сушко Д.В.

«\_\_» \_\_\_\_\_ 2022 р.

1) Назва та галузь застосування

Розробка глибокої нейромережевої системи для класифікації медичних зображень

2) Підстава для розробки

Розробку системи здійснювати на підставі наказу по університету № 203 від 14.09.2022 р. та завдання до магістерської кваліфікаційної роботи складеного та затвердженого кафедрою «Автоматизації та інтелектуальних інформаційних технологій».

3) Мета та призначення розробки

Метою даної акредитації для освітнього рівня «Магістр» є підвищення рівня цілісності подання медичної графічної інформації в умовах пандемії.

4) Джерела розробки

1. Salman F.M., Abu-Naser SS, Alajrami E, Abu-Nasser BS, Alashqar BA, COVID та SARS Detection Using Artificial Intelligence; United States: The DSpace Institutional Digital Repository System; 2020. <http://dstore.alazhar.edu.ps/xmlui/handle/123456789/587>.

2. Guo Y.R., Cao Q.D., Hong Z.S., et al. The origin, transmission and clinical therapies on corona virus disease 2019 (COVID та SARS) outbreak—an update on the status. *Mil Med Res.* 2020;7(1):1-10.

3. Worldometer. COVID та SARS CORONAVIRUS PANDEMIC. Coronavirus Cases. <https://www.worldometers.info/coronavirus/>.

4. Woolhouse M., Scott F., Hudson Z., Howey R., Chase-Topping M. Human viruses: discovery and emergence. *Philos Trans R Soc B Biol Sci.* 2012;367(1604):2864-2871.

5. Cascella M., Rajnik M., Cuomo A., Dulebohn S.C., di Napoli R. Features, evaluation and treatment corona virus (COVID та SARS). *Statpearls* [internet]. TreasureIsland (FL): StatPearls Publishing; 2020. <https://www.ncbi.nlm.nih.gov/books/NBK554776/>.

5) Показники призначення

Система для підвищення якості встановлення авторства тексту має працювати спільно з лабораторією лінгвістики Київського Національного

технічного Університету

Вихідні дані для роботи програми:

Рентгенографічне зображення легенів роздільної здатності не менше 512 x 512; кількість прихованих шарів – не більше 1000; швидкість розпізнавання не повинна бути більше 1 хв; точність розпізнавання – не менше 80%.

Результати роботи програми:

- Програмне забезпечення для класифікації по 3-м класам (нормальний, SARS, COVID) для рентгенографічного зображення.
- Прототипи моделей нейронної мережі Convolution Capsule і Visual Geometric Capsule Network.

б) Економічні показники

- прогнозовані витрати на розробку – 134 тис. грн.;
- абсолютна ефективність розробки – 2679 тис. грн.;
- термін окупності витрат для виробника – до 3-х років.

7) Стадії розробки

1. Розділ 1 «Аналіз предметної області» має бути виконаний до 15.09.2022.

2. Розділ 2 «Моделі розпізнавання вірусних захворювань легенів з використанням згорткових нейромереж» має бути виконаний до 29.09.2022.

3. Розділ 3 «Обчислювальний експеримент на основі згорткових нейромереж для задач розпізнавання медичних зображень» має бути виконаний до 19.10.2022.

4. Економічний розділ має бути виконаний до 24.11.2022.

8) Порядок контролю та приймання

1. Рубіжний контроль. Провести до 05.12.2022.
2. Попередній захист магістерської кваліфікаційної роботи. Провести до 23.12.2022.
3. Захист магістерської кваліфікаційної роботи. Провести в період з 19.12.2022 до 23.12.2022.

## Додаток Б

### Лістинг програми

```

Файл NeuralNetwork
using System;
using System.Collections.Generic; using System.Linq;
using System.Text; using System.IO;

namespace iTellUtils
{
public class Vertex
{
public List<Vertex> next; public List<double> wnext; public double delta; public double value; public double
threshold;

public Vertex()
{
next = new List<Vertex>(); wnext = new List<double>();
Random rand = new Random((int)DateTime.Now.Ticks); threshold = rand.NextDouble() * 0.4;
delta = 0.0;
value = 0.0;
}
}

public class Layer
{
public List<Vertex> vertex; public Layer(int size)
{
vertex = new List<Vertex>(); for (int i = 0; i < size; ++i)
vertex.Add(new Vertex());
}
}

public class NeuralNetworkException: Exception
{
public NeuralNetworkException(string message)
: base(message) { }
}

public class NeuralNetwork

```

```

{
private const double speed = 0.05; private List<Layer> layers;
static public NeuralNetwork Current = null; public int getDesision() {
double best = -1e100;
int bestv = -1; double[] x = Output();
for(int i=0; i<x.Length; ++i)
{
if (x[i] > best)
{
best = x[i]; bestv = i;
}
}
return bestv;
}
public double[] Output()
{
double[] result = new double[layers[layers.Count - 1].vertex.Count];

for(int i=0; i<result.Length; ++i)
{
result[i] = layers[layers.Count - 1].vertex[i].value;
}
return result;
}

public void Save(string filename)
{
StreamWriter sw = null; try
{
sw = new StreamWriter(filename);

sw.WriteLine(layers.Count);
for (int i = 0; i < layers.Count; ++i) sw.Write(layers[i].vertex.Count.ToString() + " ");
sw.WriteLine();

for (int i = 0; i < layers.Count; ++i)
for (int j = 0; j < layers[i].vertex.Count; ++j)
{
Vertex v = layers[i].vertex[j];
sw.Write(v.threshold.ToString() + " " + v.next.Count.ToString() + " "); for (int k = 0; k < v.next.Count; ++k)

```

```

{
sw.Write(v.wnext[k]);
sw.Write(" ");
}
sw.WriteLine();
}
}
catch (Exception e)
{
throw new NeuralNetworkException(e.Message);
}
finally
{
if (sw != null)
{
sw.Flush();
sw.Close();
}
}
}

public void Load(string filename)
{
StreamReader sw = null; try
{
sw = new StreamReader(filename); string s = sw.ReadToEnd();
string[] _ss = s.Split(' ', '\n', '\r'); List<string> ss = new List<string>(); for (int i = 0; i < _ss.Length; ++i)
if (_ss[i].Length > 0) ss.Add(_ss[i]);

int p = 0;
int n = int.Parse(ss[p++]); int[] size = new int[n];
for (int i = 0; i < n; ++i) size[i] = int.Parse(ss[p++]);

layers = new List<Layer>(); for (int i = 0; i < n; ++i)
layers.Add(new Layer(size[i]));

for (int i = 0; i < n - 1; ++i)
for (int j = 0; j < layers[i].vertex.Count; ++j)
for (int k = 0; k < layers[i + 1].vertex.Count; ++k) layers[i].vertex[j].next.Add(layers[i + 1].vertex[k]);

for (int i = 0; i < n; ++i)

```

```

    {
    for (int j = 0; j < size[i]; ++j)
    {
    Vertex v = layers[i].vertex[j]; v.threshold = double.Parse(ss[p++]); int m = int.Parse(ss[p++]);
    for (int k = 0; k < m; ++k) v.wnext.Add(double.Parse(ss[p++]));
    }
    }
    }
    catch (Exception e)
    {
    throw new NeuralNetworkException(e.Message);
    }
    finally
    {
    if (sw != null) sw.Close();
    }
    }

    public NeuralNetwork(string filename)
    {
    Load(filename);
    }

    public NeuralNetwork(int[] sizes)
    {
    Random rand = new Random((int)DateTime.Now.Ticks); layers = new List<Layer>();
    for (int i = 0; i < sizes.Length; ++i) layers.Add(new Layer(sizes[i]));

    for (int i = 0; i < sizes.Length - 1; ++i)
    {
    for (int j = 0; j < layers[i].vertex.Count; ++j)
    for (int k = 0; k < layers[i + 1].vertex.Count; ++k)
    {
    layers[i].vertex[j].next.Add(layers[i + 1].vertex[k]); layers[i].vertex[j].wnext.Add(rand.NextDouble() * 0.4 -
0.2);
    }
    }
    }
}

```



```

public void clear()
{
for (int i = 0; i < layers.Count; ++i)
for (int j = 0; j < layers[i].vertex.Count; ++j) layers[i].vertex[j].value = 0.0;

}

private double sigmoida(double x)
{
return 1.0 / (1.0 + Math.Exp(-x));
}

public int InputLayerSize { get { return layers[0].vertex.Count; } } public void Run(double[] input)
{
clear();
if (input.Length != layers[0].vertex.Count) throw new Exception("Input layer and input has different number of
items");

// init first layer
for (int i = 0; i < input.Length; ++i)
{
layers[0].vertex[i].value = input[i];
for (int j = 0; j < layers[0].vertex[i].next.Count; ++j) layers[0].vertex[i].next[j].value += input[i] *
layers[0].vertex[i].wnext[j];
}

// next layers
for (int t = 1; t < layers.Count; ++t)
for (int i = 0; i < layers[t].vertex.Count; ++i)
{

Vertex v = layers[t].vertex[i];
v.value = sigmoida(v.value - v.threshold); for (int j = 0; j < v.next.Count; ++j)
v.next[j].value += v.value * v.wnext[j];
}
}

public void Learn(double[] input, double[] need)
{
if (need.Length != layers[layers.Count - 1].vertex.Count)

```

```

throw new Exception("Test is something different from expected"); Run(input);
// last layer:
for (int i = 0; i < layers[layers.Count - 1].vertex.Count; ++i)
{
    Vertex v = layers[layers.Count - 1].vertex[i];
    v.delta = v.value * (1.0 - v.value) * (need[i] - v.value);
}
// other layers:
for (int t = layers.Count - 2; t >= 0; --t) foreach (Vertex v in layers[t].vertex)
{
    v.delta = 0.0;
    for (int i = 0; i < v.next.Count; ++i) v.delta += v.next[i].delta * v.wnext[i];
    v.delta *= v.value * (1.0 - v.value);
}

// learning:

for (int t = layers.Count - 1; t >= 0; --t)
for (int i = 0; i < layers[t].vertex.Count; ++i)
{
    Vertex v = layers[t].vertex[i];
    for (int j = 0; j < v.next.Count; ++j)
    {
        Vertex y = v.next[j];
        v.wnext[j] += speed * y.delta * v.value;
    }
}
}
}

Файл WordDB
using System;
using System.Collections.Generic; using System.Linq;
using System.Text; using System.IO;
using System.Runtime.Serialization.Formatters.Binary; using System.Runtime.Serialization;

namespace iTellUtils
{

public class WordDBCantSaveException: Exception

```

```

{
public WordDBCantSaveException(string s):base(s) { }
}
public class WordDBCantLoadException : Exception
{
public WordDBCantLoadException(string s) : base(s) { }
}

[Serializable] public class WordDB
{
static private WordDB instance = null;
static public WordDB Instance { get { if (instance == null) instance = new WordDB(); return instance; } }

private SortedSet<string> words = new SortedSet<string>();

private double getProb(string s, FuzzyVector v) { if (s.Length != v.chars.Count) return 0.0; double prob = 1.0;
for(int i=0; i<s.Length; ++i)
prob *= v.chars[i][(int)(s[i]-'a')]; return prob;
}

public int WordsCount { get { return words.Count; } } private bool checkPref(string s1, string s2)
{
if (s1.Length<s2.Length) return false;
int i = 0; while (i < s2.Length) if (s1[i] != s2[i]) return false; else ++i; return true;
}
public string[] getWords(string pref)
{
string[] w = words.ToArray(); List<string> list = new List<string>(); for (int i = 0; i < w.Length; ++i)
{
if (checkPref(w[i], pref)) list.Add(w[i]);
}
return list.ToArray();
}

public void LoadFromTextFile(string filename)
{
try
{

```

```

words.Clear();
string[] w = File.ReadAllLines(filename);
for (int i = 0; i < w.Length; ++i) if (w[i].Length > 0)
{

AddWord(w[i]);
}
}
catch (Exception e)
{
throw new WordDBCantLoadException(e.Message);
}
}
public void SaveToTextFile(string filename)
{

try
{

}

string[] s = words.ToArray(); File.WriteAllLines(filename, s);

catch (Exception e)
{
throw new WordDBCantSaveException(e.Message);
}
}

public WordDB() { }
public WordDB(string[] words)
{
for (int i = 0; i < words.Length; ++i) this.words.Add(words[i]);
}
public void AddWord(string s)
{
string ss="";
for (int i = 0; i < s.Length; ++i) ss += Char.ToLower(s[i]); if (words.Contains(ss)) return;
words.Add(ss);
}

```

```

}
public void RemoveWord(string s)
{
if (!words.Contains(s)) return; words.Remove(s);
}

public WordDBSearchResult findBestWords(FuzzyVector word)
{

WordDBSearchResult res = new WordDBSearchResult(10); string[] W = getWords("");
for (int i = 0; i < W.Length; ++i)
{
res.AddWord(W[i], getProb(W[i], word));
}
return res;
}
}

public class WordDBSearchResult
{
private List<double> prob; private List<string> word;
public WordDBSearchResult(int limit) { prob = new List<double>();
word = new List<string>();
}

public void AddWord(string word, double prob)
{
this.prob.Add(prob); this.word.Add(word);
}
public int Count()
{return 0;}

public string GetFirstWord()
{
double best = 0.0; string res = "";
for (int i=0; i < prob.Count; ++i)
{
if (prob[i] > best)
{
best = prob[i]; res = word[i];
}
}
}
}

```

```

    }
    }
    return res;
    }
    public string GetWordAt(int index)
    { return null; }
    public double GetProbAt(int index)
    { return 0.0; }
    }

```

```

public class FuzzyVector
{
    public List<double[]> chars = new List<double[]>();
}
}

```

Файл BlockRecognizer

```

using System;
using System.Collections.Generic; using System.Linq;
using System.Text; using System.Drawing;

```

```

namespace iTellUtils

```

```

{
    public class BlockRecognizer
    {
        static private int Width; static private int Height; static int[,] bits = null;

```

```

        static public bool Loaded() { return (bits != null); } static public void StartImage(int [,] img, int W, int H)
        {
            Width = W; Height = H; bits = img; line = 0;
            column = 0;
        }

```

```

        static public void Reset()
        {
            line = column = 0;
        }

```

```

        public static int dx0 = 2; public static int dy0 = 0; public static int dx = 7; public static int dy = 16;

```

```
static public int line; static private int column;
```

```
static public Bitmap NextBitmapChar()
```

```
{
if ((line+1)*dy + dy0 > Height) return null; int y = line * dy + dy0;
int x = column * dx + dx0; if (x + dx > Width) {
++line; column = 0;
return NextBitmapChar();
}
Bitmap res = new Bitmap(dx, dy); int sum = 0;
for (int i = 0; i < dx; ++i) for (int j = 0; j < dy; ++j) {
//res.SetPixel(i, j, Color.Red); sum += bits[x + i, y + j];
res.SetPixel(i, j, Color.FromArgb(255 - bits[x + i, y + j], 255 - bits[x + i, y + j], 255
- bits[x + i, y + j]));
}
++column; return res;
}
```

```
static public double[] NextDoubleChar()
```

```
{
if ((line + 1) * dy + dy0 > Height) return null; int y = line * dy + dy0;
int x = column * dx + dx0; if (x + dx > Width)
{
++line; column = 0;
return NextDoubleChar();
}
double[] res = new double[dx*dy]; int sum = 0;
for (int j = 0; j < dy; ++j) for (int i = 0; i < dx; ++i)
{
res[sum++] = (double)bits[x + i, y + j]/255.0;
}
++column; return res;
}
}
}
```

Файл ImageFilter

```
using System;
using System.Collections.Generic; using System.Linq;
using System.Text; using System.Drawing;
```

```

namespace iTellUtils
{
public class ImageFilter
{
static int ColorToInt(Color v)
{
return Math.Max(v.B,Math.Max(v.G,v.R));
}
static Color IntToColor(int x) {

return Color.FromArgb(x, x, x);
}

public static Bitmap DoFilteringSimple(Bitmap source) { Bitmap result = (Bitmap)source.Clone();
for (int x = 0; x < result.Width; ++x)
for (int y = 0; y < result.Height; ++y)
{
result.SetPixel(x,y,IntToColor(ColorToInt(source.GetPixel(x, y))));
}
return result;
}

public static Bitmap DoFiltering(Bitmap source)
{
Bitmap result = new Bitmap(source.Width, source.Height); int[] pixels = new int[8];
int[] dx = {-1,-1,-1,0,1,1, 1, 0};
int[] dy = {-1, 0, 1,1,1,0,-1,-1};
for (int x = 0; x < result.Width; ++x)
{
for (int y = 0; y < result.Height; ++y)
{
if (x == 317 && y == 106) { Console.WriteLine();
}
int n = 0;
for(int i=0; i<8; ++i)
{
int xx=x+dx[i]; int yy=y+dy[i];
if (xx<0 || yy<0 || xx>=source.Width || yy>=source.Height) continue; pixels[n++] =
ColorToInt(source.GetPixel(xx,yy));
}
}
}
}
}

```



```

    }
    for(int i=0; i<n-1; ++i) for(int j=0; j<n-1; ++j)
    if (pixels[j]>pixels[j+1])
    {

    }
    if (n<8) {

    pixels[j]^=pixels[j+1]; pixels[j+1]^=pixels[j]; pixels[j]^=pixels[j+1];

    result.SetPixel(x,y,IntToColor(ColorToInt(source.GetPixel(x,y))));
    }
    else
    {
    int X = ColorToInt(source.GetPixel(x,y));
    double med = (double)(pixels[3]+pixels[4])/2.0d; double r1 = ((double)X<=med)?pixels[0]-X:X-pixels[7];
    double r2 = ((double)X<=med)?pixels[1]-X:X-pixels[6]; double r3 = ((double)X<=med)?pixels[2]-X:X-pixels[5];
    double r4 = ((double)X<=med)?pixels[3]-X:X-pixels[4];

    if (r1>8.0 || r2>20.0 || r3>40.0 || r4>80.0)
    {

    }
    else
    {

    }
    }
    }

    result.SetPixel(x,y,IntToColor( (int)(1.0*med + 0.0*X + 0.5) ));

    result.SetPixel(x,y,IntToColor( (int)(0.0*med + 1.0*X + 0.5) ));

    return result;
}

```

```

}
}

```

Файл Recognizer

```

using System;
using System.Collections.Generic; using System.Linq;
using System.Text; using iTellUtils;

```

```

using System.Drawing;
using System.Drawing.Imaging;

```

```

namespace iTell

```

```

{
public class Recognizer
{
static public List<double[]> getData(int[][] images)
{
List<double[]> res = new List<double[]>(); for (int i = 0; i < images.Length; ++i)
{
double[] a = new double[NeuralNetwork.Current.InputLayerSize]; for (int r = 0; r < images[i].Length; ++r)
a[r] = (double)images[i][r] / 256.0; res.Add(a);
}
return res;
}
}

```

```

static public FuzzyVector GetVector(List<double[]> data)
{
FuzzyVector vect = new FuzzyVector(); for (int i = 0; i < data.Count; ++i) {
NeuralNetwork.Current.Run(data[i]); double[] v = NeuralNetwork.Current.Output(); double sum = 0.0;
for (int j = 0; j < 26; ++j) sum += v[j]; for (int j = 0; j < 26; ++j) v[j] /= sum; vect.chars.Add(v);
}
return vect;
}
}

```

```

static public FuzzyVector GetVector(string word)
{
FuzzyVector vect = new FuzzyVector(); for (int i = 0; i < word.Length; ++i)
{
double[] y = new double[26]; for (int j = 0; j < 26; ++j)
y[j]=((int)(word[i]-'a')==j)?1.0:0.0;
}
}

```

```

vect.chars.Add(y);
}
return vect;
}
}
}

```

Файл MainForm

```

using System;
using System.Collections.Generic; using System.ComponentModel; using System.Data;
using System.Drawing; using System.Linq; using System.Text;
using System.Windows.Forms; using iTellUtils;
using System.IO;

namespace iTell
{
public partial class MainForm : Form
{
private ConsoleForm consoleForm = new ConsoleForm(); public void showConsole(bool yes)
{
UserSettings.Instance.ShowConsole = yes; вывестиКонсольToolStripMenuItem.Checked = yes; if (yes)
consoleForm.Show(); else
consoleForm.Hide();
}

public MainForm()
{
InitializeComponent();
}

private void LoadWordDB(string filename)
{
Console.Message("Завантаження бази слів з файлу " + filename); bool ok = true;
try { WordDB.Instance.LoadFromTextFile(filename); }
catch (WordDBCantLoadException e) { ok = false; Console.Error("LoadWordDB", "Помилка завантаження
слів з файлу: " + e.Message); }
if (ok) Console.Message("База слів завантажена");
}

private void SaveWordDB(string filename)

```

```

    {
        Console.Message("Збереження бази слів до файлу " + filename); bool ok = true;
        try { WordDB.Instance.SaveToTextFile(filename); }
        catch (WordDBCantSaveException e) { ok = false; Console.Error("SaveWordDB", "Помилка збереження
слів до файлу: " + e.Message); }
        if (ok) Console.Message("База слів збережена");
    }

private void Form1_Load(object sender, EventArgs e)
{
    consoleForm.Width = 600;
    consoleForm.Height = 200; Console.setForm(consoleForm);
    if (UserSettings.Instance.ShowConsole)
    {
        consoleForm.Show(); вивестиКонсольToolStripMenuItem.Checked = true;
    }
    else
    {
        consoleForm.Hide(); вивестиКонсольToolStripMenuItem.Checked = false;
    }

    // todo loading words
    if (UserSettings.Instance.CurrentDB != "") LoadWordDB(UserSettings.Instance.CurrentDB);
    // todo loading net
    if (UserSettings.Instance.CurrentNetwork != "")
        NeuralNetwork.Current = new NeuralNetwork(UserSettings.Instance.CurrentNetwork); else
    {
        int[] sizes = new int[] { 112,100,26 }; NeuralNetwork.Current = new NeuralNetwork(sizes);
    }
}

private void button1_Click(object sender, EventArgs e)
{
}

private void button2_Click(object sender, EventArgs e)
{
}

private void button1_Click_1(object sender, EventArgs e)

```

```

{
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
pictureBox1.Load(openFileDialog1.FileName);
Bitmap bits = ImageFilter.DoFilteringSimple(new Bitmap(pictureBox1.Image)); pictureBox1.Image = bits;
int [,] img = new int[bits.Width, bits.Height]; for (int i = 0; i < bits.Width; ++i)
for (int j = 0; j < bits.Height; ++j)
{
img[i, j] = 255 - (int)bits.GetPixel(i, j).B;

}
BlockRecognizer.StartImage(img, bits.Width, bits.Height);
}
}

string goWork(FuzzyVector v, List<double[]> input)
{
string s = WordDB.Instance.findBestWords(v).GetFirstWord(); for (int i = 0; i < s.Length; ++i) {
int a = (int)s[i] - 'a'; double[] need = new double[26];
for (int j = 0; j < s.Length; ++j)
{
need[j] = (j == a) ? 1.0 : 0.0;
}
NeuralNetwork.Current.Learn(input[i], need);
}
return s;
}

private void button2_Click_1(object sender, EventArgs e)
{
DateTime beginTime = DateTime.Now; int total=0; BlockRecognizer.Reset();
FuzzyVector word = new FuzzyVector(); List<double[]> imgs = new List<double[]>(); StringBuilder sb = new
StringBuilder(); int lastline = BlockRecognizer.line;
int last = 0; while(true) {
last = BlockRecognizer.line;
double[] x = BlockRecognizer.NextDoubleChar();
++total;
if (x == null) break;
int spaces = 0; for (int i = 0; i < x.Length; ++i) if (x[i] < 0.1) spaces++; if (spaces > x.Length * 90 / 100)
{
if (word.chars.Count > 0)

```

```

{
if (last != lastline)
{
sb.Append("\n");
}

}
}
else
{

lastline=last; sb.Append(goWork(word, imgs)+" "); word.chars.Clear();
imgs.Clear();

imgs.Add(x); NeuralNetwork.Current.Run(x);
word.chars.Add(NeuralNetwork.Current.Output());
}
}
if (word.chars.Count > 0)
{
sb.Append(goWork(word, imgs)); word.chars.Clear(); imgs.Clear();
}
richTextBox1.Text = sb.ToString();
double v=(DateTime.Now - beginTime).TotalSeconds; MessageBox.Show("Затрачений час: " + v.ToString()
+ " сек.\n" +
"Всього символів: " + total.ToString() + "\n" +
"Час обробки одного символу: " + (v / (double)total).ToString() + "
сек/символ");
}

private void вивестиКонсольToolStripMenuItem_Click(object sender, EventArgs e)
{
showConsole(!UserSettings.Instance.ShowConsole);
}

private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
{
Close();
}

private void налаштуванняToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
SettingsForm settingsForm=new SettingsForm(this); settingsForm.ShowDialog();
}

private void редагуванняБазиToolStripMenuItem_Click(object sender, EventArgs e)
{
DBForm form = new DBForm(); form.ShowDialog();
}

private void імпортуватиЗФайлуToolStripMenuItem_Click(object sender, EventArgs e)
{
if (openFileDialog1.ShowDialog() == DialogResult.OK) LoadWordDB(openFileDialog1.FileName);
}

private void експортуватиВФайлToolStripMenuItem_Click(object sender, EventArgs e)
{
if (saveFileDialog1.ShowDialog()==DialogResult.OK) SaveWordDB(saveFileDialog1.FileName);
}

private void button4_Click(object sender, EventArgs e)
{
if (BlockRecognizer.Loaded()) { TestForm form = new TestForm(); form.ShowDialog();
}
}

private void завантажитиНейроннуМережуToolStripMenuItem_Click(object sender, EventArgs e)
{
if (openFileDialog1.ShowDialog() == DialogResult.OK)
NeuralNetwork.Current.Load(openFileDialog1.FileName);
}

private void зберегтиНейроннуМережуToolStripMenuItem_Click(object sender, EventArgs e)
{
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
NeuralNetwork.Current.Save(saveFileDialog1.FileName);
}

private void навчанняНаОсновіШаблоннихСимволівToolStripMenuItem_Click(object sender, EventArgs e)
{
LearningForm form = new LearningForm(); form.ShowDialog();
}

```

```

    }

    private void button3_Click(object sender, EventArgs e)
    {
        if (saveFileDialog1.ShowDialog() == DialogResult.OK) {
            File.WriteAllText(saveFileDialog1.FileName, richTextBox1.Text);
        }
    }

    private void завантажитиЗображенняToolStripMenuItem_Click(object sender, EventArgs e)
    {
        button1_Click_1(sender, e);
    }

    private void зберегтиРезультатиРозпізнаванняToolStripMenuItem_Click(object sender, EventArgs e)
    {
        button3_Click(sender, e);
    }
}

Файл LearningForm
using System;
using System.Collections.Generic; using System.ComponentModel; using System.Data;
using System.Drawing; using System.Linq; using System.Text;
using System.Windows.Forms; using iTellUtils;

namespace iTell
{
    public partial class LearningForm : Form
    {
        public LearningForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
            {
                textBox1.Text = folderBrowserDialog1.SelectedPath;
            }
        }
    }
}

```



```

}
}

```

e) Random rand = new Random((int)DateTime.Now.Ticks); private void button2\_Click(object sender, EventArgs

```

{
List<double[]> bits = new List<double[]>(); List<double[]> need = new List<double[]>(); for(int i=0; i<26;
++i)

```

```

{
double[] res = new double[7*16];
Bitmap p = new Bitmap(textBox1.Text+"\"+(char)((int)'A'+i)+".bmp"); int pos=0;
for(int y=0; y<16; ++y) for(int x=0; x<7; ++x) res[pos++] = (double)(255- p.GetPixel(x,y).B)/255.0;
bits.Add(res);
}

```

```

for(int i=0; i<26; ++i)
{
double[] a = new double[26];
for(int j=0; j<26; ++j) a[j] = (i==j)?1.0:0.0; need.Add(a);
}

```

```

StringBuilder sb = new StringBuilder(); int iter = (int)numericUpDown1.Value; int total=0;
int wins=0; progressBar1.Minimum=0; progressBar1.Maximum=iter;
for (int i = 0; i < iter; ++i)
{
progressBar1.Value = i; Application.DoEvents();
for (int a = 0; a < 100; ++a)
{
int x = rand.Next() % 26;

```

```

NeuralNetwork.Current.Run(bits[x]);
if (NeuralNetwork.Current.getDesision()==x) {sb.Append("+");++wins;} else sb.Append("o");
++total;
NeuralNetwork.Current.Learn(bits[x], need[x]);
}
progressBar1.Value = i + 1;
}
richTextBox1.Text = sb.ToString() + "\n"+"["+wins.ToString()+"] of "+total.ToString()+"]";
}

```

```

private void button4_Click(object sender, EventArgs e)

```

```
{  
    if (openFileDialog1.ShowDialog() == DialogResult.OK)  
        NeuralNetwork.Current.Load(openFileDialog1.FileName);  
}
```

```
private void button3_Click(object sender, EventArgs e)  
{  
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)  
        NeuralNetwork.Current.Save(saveFileDialog1.FileName);  
}
```

## Додаток В. Ілюстративна частина

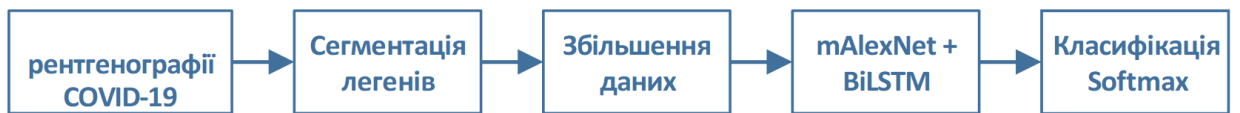


Рис. 1 Схема аналітичного опрацювання візуальних даних

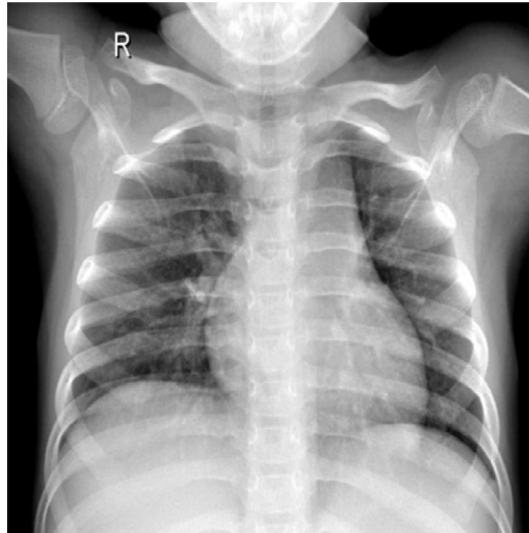


Рис. 2 Зразок рентгенографічного зображення легенів здорової особи



Рис. 3 Зразок рентгенографічного зображення легенів інфікованого COVID



Рис. 4 Зразок рентгенографічного зображення легенів інфікованого вірусною пневмонією пацієнта

Автор	Метод	Точність, %
1	2	3
Укар та Коркмаз [36]	Bayes-SqueezeNet	98.30
Озтарк та ін. [37]	DarkCovidNet	87.02
1	2	3
Ванг та Вонг [43]	ББ	92.30
Гупта та ін. [44]	In1aCovNet1-19	99.08
Ху та ін. [45]	Кев№1 + Фокус уваги	86.70
Сеті та Бехера [46]	Кев№150 + БУМ	95.40
Хемдан, Шоуман та Карар [47]	УОП19	90.00
Рахімзаде та Агтар [48]	ХеерНоп + Кев№150У2	91.40
Аслан та ін. [33]	Гібридний метод	98.70
Афшар та ін. [49]	Капсульні мережі	95.70
Ноур, Кьомерт та Полат [31]	Машинне ББ	98.97
Апостолопулос та Мпесьяна [50]	Передача навчання	93.48
Кхан, Шан та Бхат [38]	Передача навчання	95.00
Язіф та Венхай [34]	Передача навчання	96.00
Фарук та Хафіз [51]	Передача навчання	96.20
Чоударі та ін. [32]	Передача навчання	97.94
Нарін, Кая та Памук [40]	Передача навчання	98.00

Рис. 5 Порівняння методів аналітичного опрацювання медичних зображень

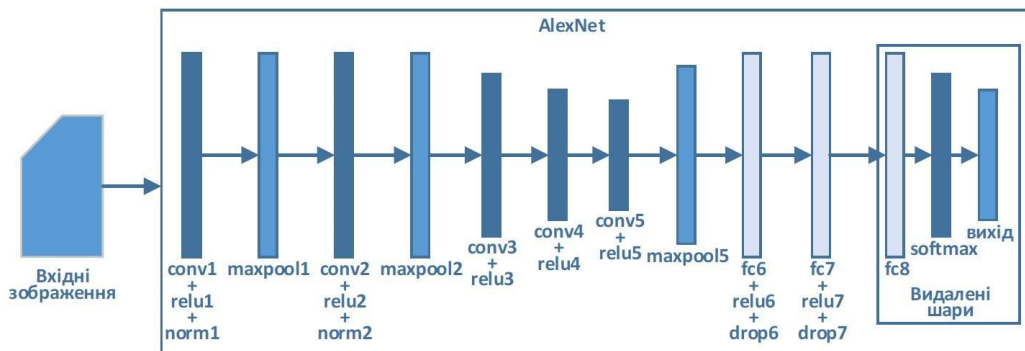


Рис. 6 Архітектура AlexNet

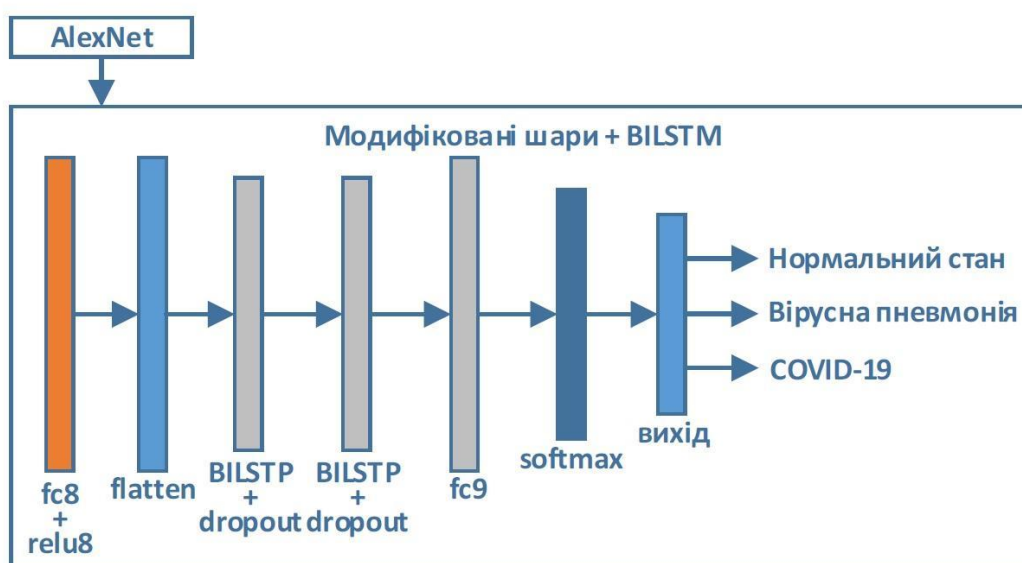


Рис. 7 Гібридна архітектура mAlexNet-BiLSTM

Назва шару	Розмір	Розмір фільтра	Вкладення	Прокладка	Вихідний канал	Функція активації
conv1	55 x 55	11 x 11	4	0	96	relu
maxpool1	27 x 27	3 x 3	2	0	96	—
conv2	27 x 27	5 x 5	1	2	256	relu
maxpool2	13 x 13	3 x 3	2	0	256	—
conv3	13 x 13	3 x 3	1	1	384	relu
conv4	13 x 13	3 x 3	1	1	384	relu
conv5	13 x 13	3 x 3	1	1	256	relu
maxpool5	6 x 6	3 x 3	2	0	256	—
1	2	3	4	5	6	7
fc6	—	—	—	—	4096	relu
fc7	—	—	—	—	4096	relu
fc8	-	—	—	—	25	relu
fc9	-	—	—	—	2	softmax

Рис. 8 Характеристики шарів mAlexNet

Оптимізатор	Макс. епох	Мін. партія	Початкова швидкість навчання ( $\alpha$ )	Імпульс ( $\gamma$ )
SGDM	100	60	0.001	0.95

Рис. 9 Параметри тренування mAlexNet

BiLSTM-1			BiLSTM-2			fc9	
Кількість прихованих шарів	Функція активації		Кількість прихованих шарів	Функція активації		Вихідний розмір	State функція активації
	State	Gate		State	Gate		
125	<i>tanh</i>	<i>sigmoid</i>	100	<i>tanh</i>	<i>sigmoid</i>	3	<i>tanh</i>

BiLSTM-1			BiLSTM-2			fc9	
Кількість прихованих шарів	Функція активації		Кількість прихованих шарів	Функція активації		Вихідний розмір	State функція активації
	State	Gate		State	Gate		
125	<i>tanh</i>	<i>sigmoid</i>	100	<i>tanh</i>	<i>sigmoid</i>	3	<i>tanh</i>

Рис. 10 Гіперпараметри шарів гібридної архітектури

Оптимізатор	Коефіцієнт градієнтного розпаду ( $\beta_1$ )	Квадратичний коефіцієнт розпаду градієнта ( $\beta_2$ )	Макс. епох	Мін. розмір партії	Початкова швидкість навчання ( $\alpha$ )	Епсилон ( $\epsilon$ )
Adam	0.9	0.999	200	512	0.001	10 <sup>-8</sup>

Рис. 11 Тренувальні параметри гібридної архітектури

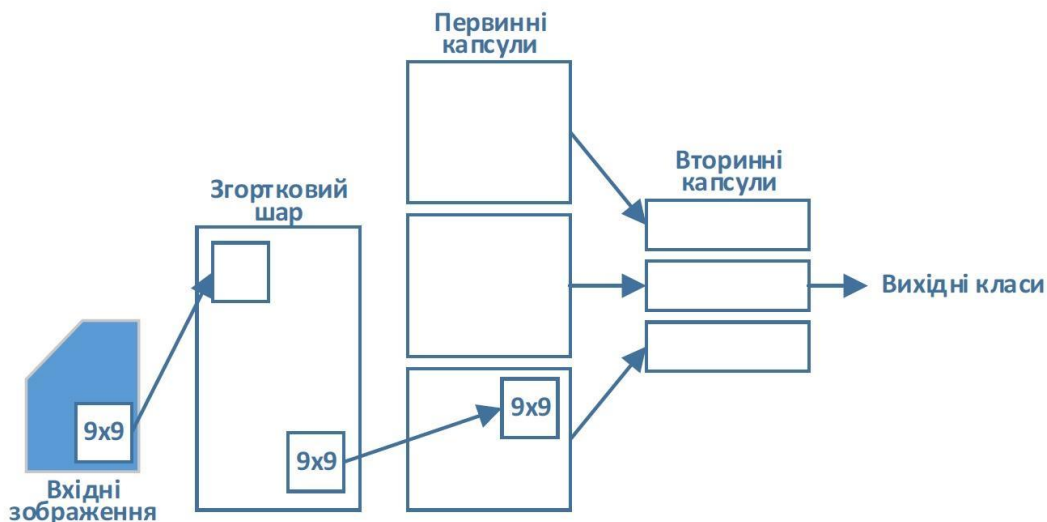


Рис. 12 Архітектура CapsNet для виявлення COVID та SARS

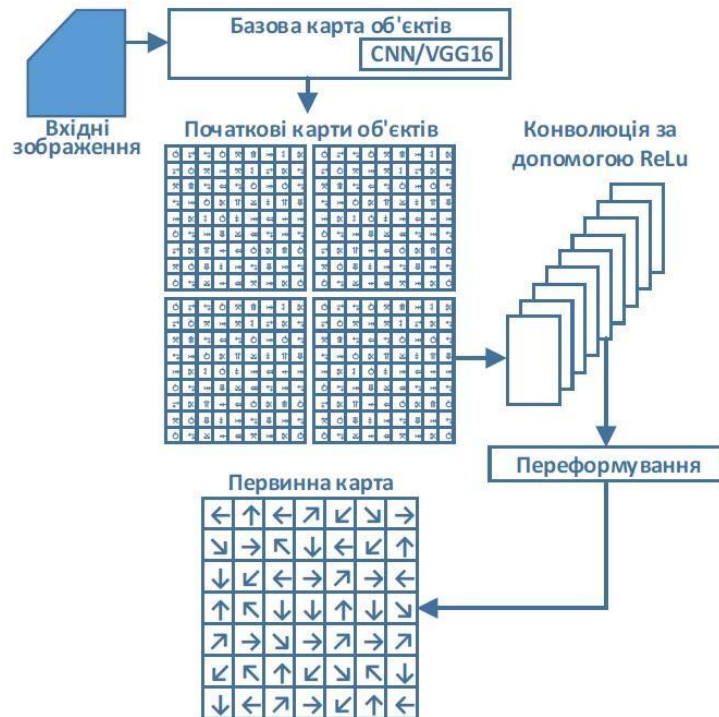


Рис. 13 Базова архітектура моделі VGG- CapsNet для виявлення COVID та SARS

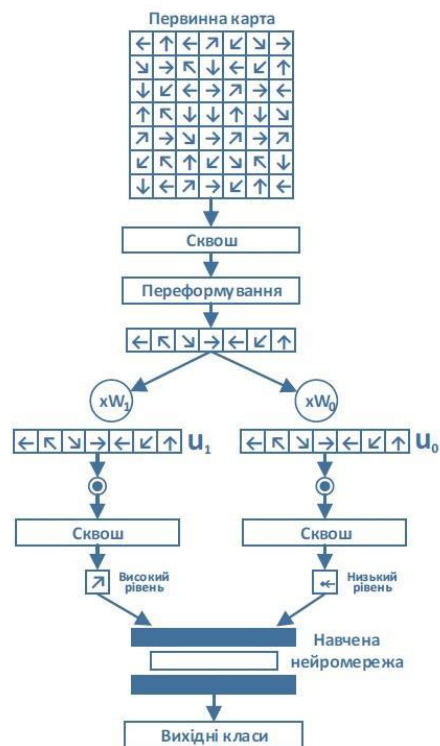


Рис. 14 Узагальнююча складова архітектури моделі VGG- CapsNet для виявлення COVID та SARS

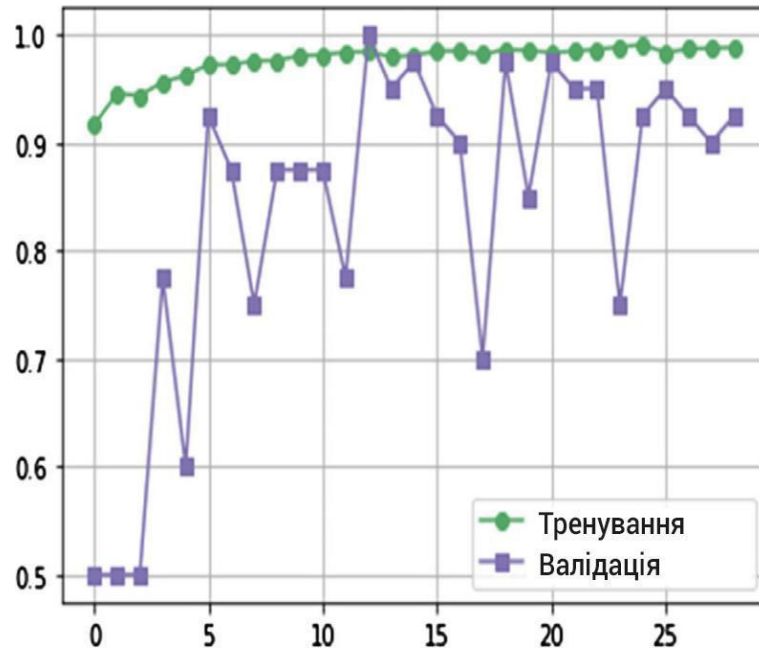


Рис. 15 Точності моделі CNN-CapsNet для задачі бінарної класифікації

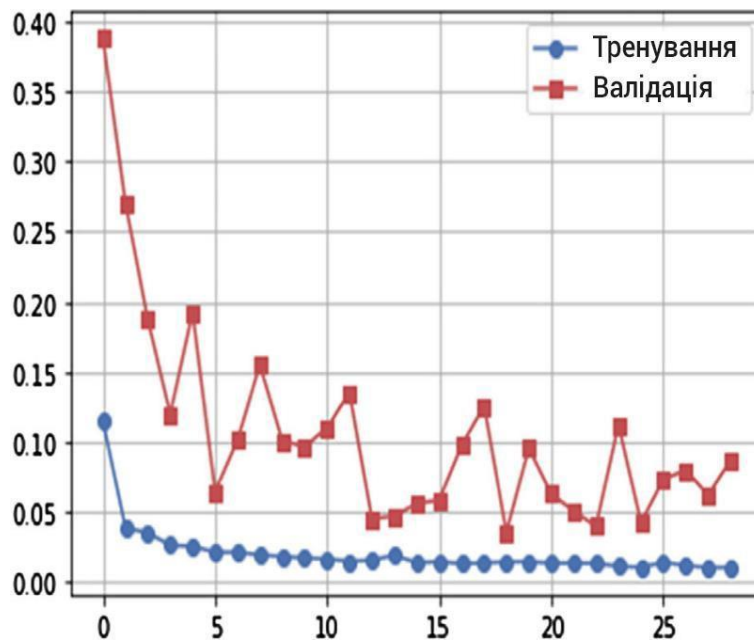


Рис. 16 Втрати моделі CNN-CapsNet для задачі бінарної класифікації



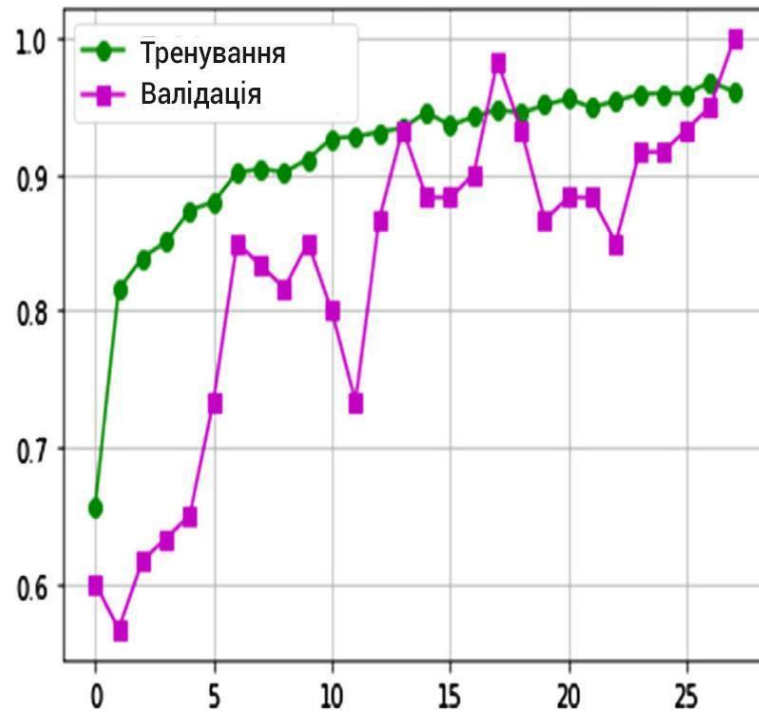


Рис. 17 Точність моделі CNN-CapsNet для трикласової задачі

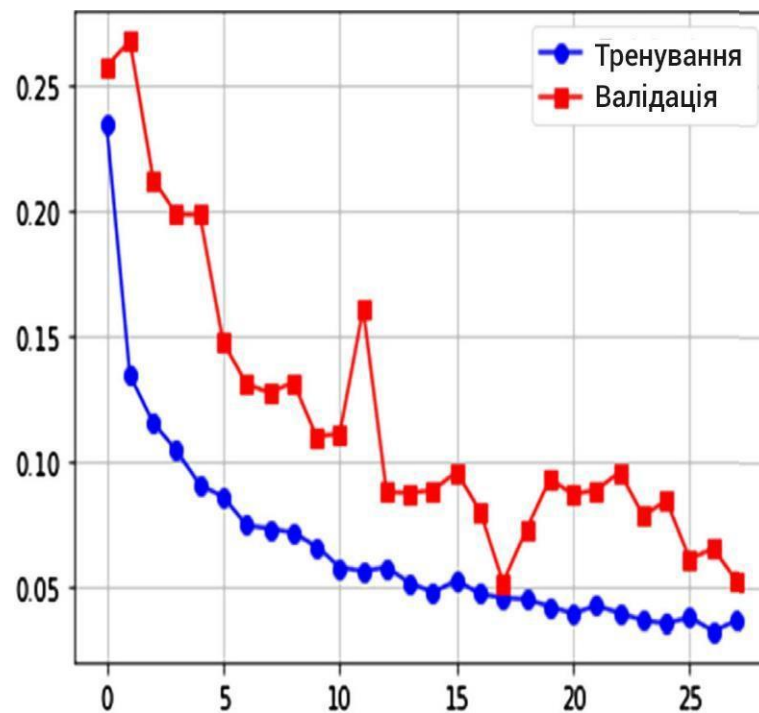


Рис. 18 Втрати моделі CNN-CapsNet для трикласової задачі

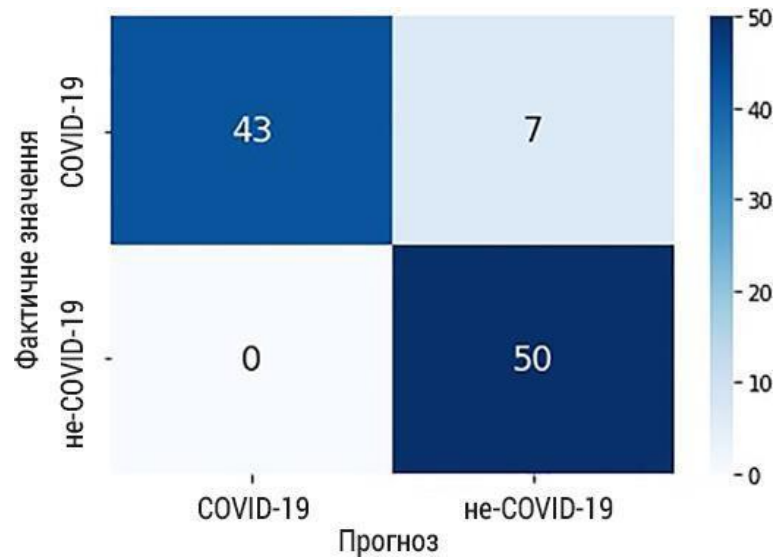


Рис. 19 Матриця сплутування моделі CNN-CapsNet для двокласової задачі

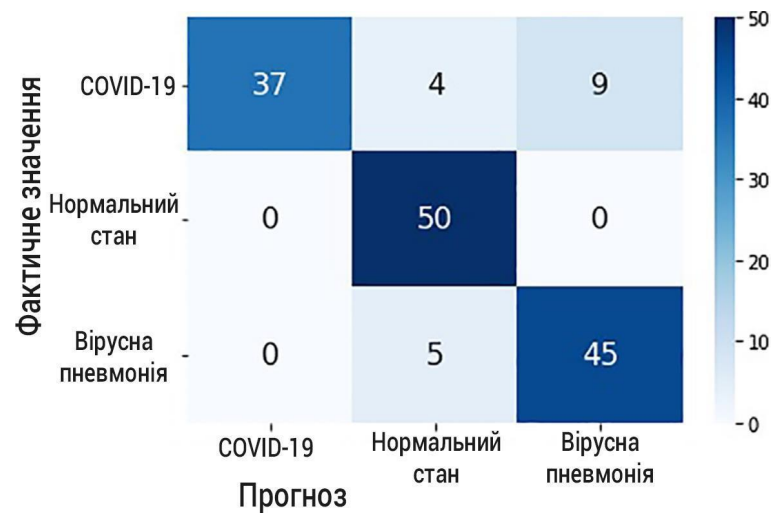


Рис. 20 Матриця сплутування моделі CNN-CapsNet для трикласової задачі

Клас	CNN-CapsNet			VGG-CapsNet		
	Точність	Запам'ятовуваність	Оцінка F1	Точність	Запам'ятовуваність	Оцінка F1
«COVID та CAPS»	1.00	0.90	0.95	0.96	0.98	0.97
«Не-COVID19»	0.91	1.00	0.95	0.98	0.96	0.97
Точність	0.95			0.97		

Рис. 21 Результати ефективності моделей CNN-CapsNet і VGG-CapsNet при вирішенні задачі бінарної класифікації

Клас	CNN-CapsNet			VGG-CapsNet		
	Точність	Запам'ятовуваність	Оцінка F1	Точність	Запам'ятовуваність	Оцінка F1
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
«COVID та SARS»	0.96	0.86	0.91	0.96	0.98	0.97
Звичайна пневмонія	0.89	1.00	0.94	0.87	0.96	0.91
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
Вірусна пневмонія	0.88	0.86	0.87	0.92	0.82	0.87
Точність	0.91			0.92		

Рис. 22 Показники продуктивності результатів експерименту для моделей CNN-CapsNet і VGG-CapsNet для трикласових задач

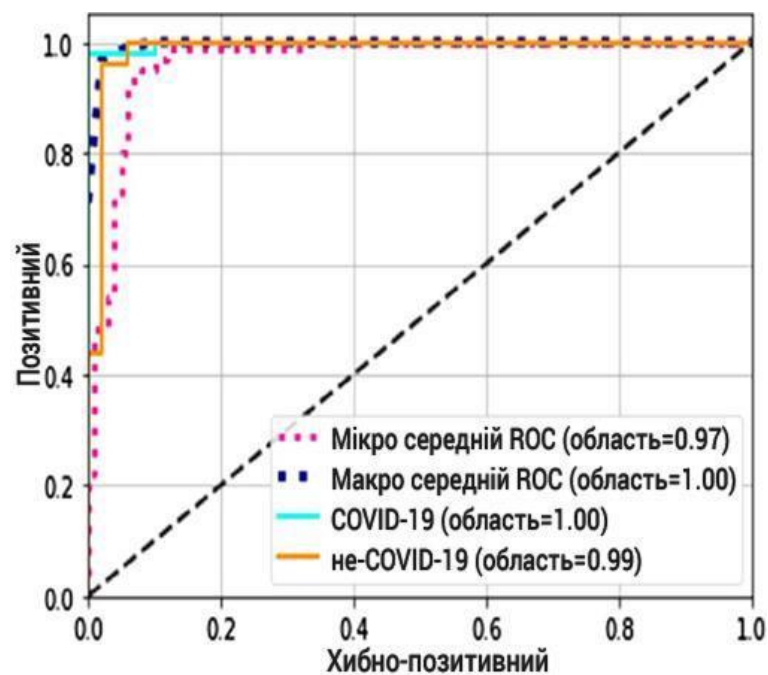


Рис. 23 Характеристики моделі CNN-CapsNet для задачі бінарної класифікації

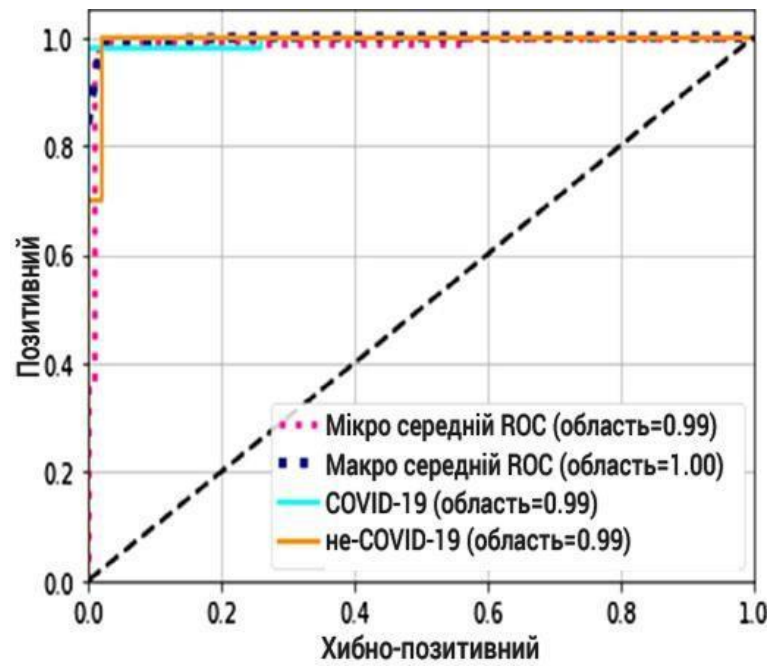


Рис. 24 Характеристики моделі VGG-CapsNet для задачі бінарної класифікації

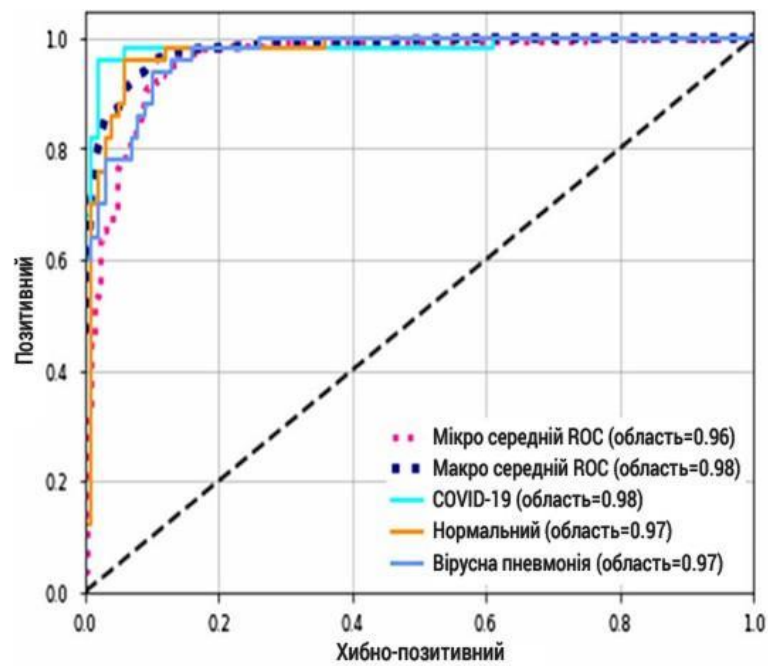


Рис. 25 Характеристики моделі CNNCapsNet для класифікації на три класи

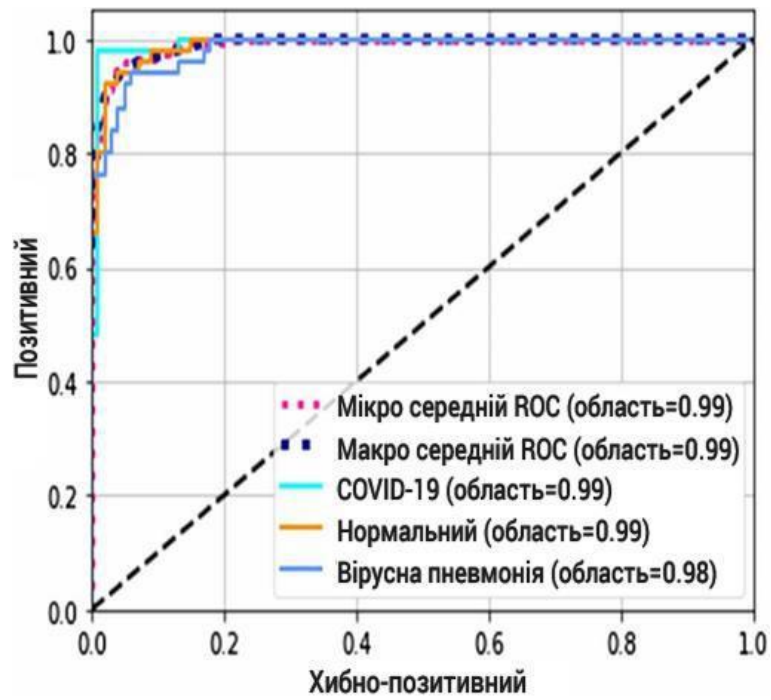


Рис. 26 Характеристики моделі VGG-CapsNet для класифікації на три класи

## ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: «РОЗРОБКА ГЛИБОКОЇ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ ДЛЯ КЛАСИФІКАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ»

Тип роботи: кваліфікаційна робота

(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше – зазначити)

Підрозділ: кафедра АІТ, ФКСА, 1АКІТ-21м

(кафедра, факультет, навчальна група)

Науковий керівник: Кулик Я.А., доц. каф. АІТ

(прізвище, ініціали, посада)

### Показники звіту подібності

<i>Plagiat.pl (StrikePlagiarism)</i>		<i>Unicheck</i>	
КП1		Оригінальність	
КП2			
Тривога/Білі знаки		Схожість	

Аналіз звіту подібності (відмітити потрібне)

**X Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор \_\_\_\_\_ Сушко Д.В.  
(підпис) (прізвище, ініціали)

Опис прийнятого рішення:

Допустити до захисту

Особа, відповідальна за перевірку \_\_\_\_\_ Маслій Р.В.  
(підпис) (прізвище, ініціали)