

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**  
Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_ Ю. П. Кондратенко  
«\_\_\_» \_\_\_\_\_ 2022 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

**ДІАГНОСТУВАННЯ COVID-19 НА ОСНОВІ МЕТОДІВ**  
**МАШИННОГО НАВЧАННЯ**

Спеціальність 124 «Системний аналіз»

**124 – МКР – 607.21610120**

Студент \_\_\_\_\_ О.А. Савчук  
«\_\_\_» \_\_\_\_\_ 2022 р.

Консультант \_\_\_\_\_ Ю.П. Кондратенко  
докт. техн. наук, проф.  
«\_\_\_» \_\_\_\_\_ 2022 р.

**Миколаїв – 2022**

**Чорноморський національний університет ім. Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

Освітньо-кваліфікаційний рівень **магістр**

Галузь знань **12 «Інформаційні технології»**

*(шифр і назва)*

Спеціальність **124 «Системний аналіз»**

*(шифр і назва)*

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.

\_\_\_\_\_ Ю. П. Кондратенко

«    » \_\_\_\_\_ **2022р.**

**ЗАВДАННЯ**

**на магістерську кваліфікаційну роботу**

**Савчуку Олексію Андрійовичу**

1. Тема магістерської кваліфікаційної роботи «Діагностування COVID-19 на основі методів машинного навчання».

Керівник роботи Кондратенко Юрій Пантелійович, д-р техн. наук, професор.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «\_\_» \_\_\_\_ 2021 р. № \_\_\_\_\_

2. Строк подання студентом роботи \_\_\_\_ \_\_\_\_\_ 2022 р.

3. Вхідні (початкові) дані до роботи: графічні дані, а саме КТ знімки легень, розбиті на множини позитивних та негативних прикладів для навчання мережі.

Очікуваний результат: система діагностування COVID-19 на основі знімків КТ для виявлення COVID-19.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- аналіз сучасного стану задачі діагностування COVID-19;
- огляд існуючих методів обробки медичних даних;
- створення математичної моделі нейронної мережі;
- порівняльний аналіз результатів застосування обраних методів

діагностування на основі нейронних мереж для розв'язання поставленої задачі.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: аналіз умов праці та безпека у надзвичайних ситуаціях у контексті реагування на COVID-19

---

---

---

7. Консультанти розділів роботи

| Розділ                             | Прізвище, ініціали та посада консультанта | Підпис |
|------------------------------------|---|--------|
| Спеціальна частина з охорони праці | д.б.н., проф. Григор'єва Людмила Іванівна |        |
| Методична частина                  | к.т.н., доцент Сіденко Євген Вікторович   |        |

Керівник роботи д-р техн. наук, проф. Кондратенко Ю. П.  
(наук. ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Завдання прийнято до виконання Савчук О. А.  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

Дата видачі завдання «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

# КАЛЕНДАРНИЙ ПЛАН

## Виконання магістерської кваліфікаційної роботи

Тема: Діагностування COVID-19 на основі методів машинного навчання

| №  | Найменування роботи  | Початок    | Закінчення | Примітки |
|----|--|------------|------------|----------|
| 1  | Визначення керівника і теми МКР.<br>Подання заяви на затвердження теми МКР   | 01.09.2021 | 10.10.2021 |          |
| 2  | Отримання завдання на виконання МКР  | 19.10.2021 | 22.10.2021 |          |
| 3  | Складання календарного плану на період виконання МКР   | 23.10.2021 | 26.10.2021 |          |
| 4  | Огляд літератури за темою дослідження  | 27.10.2021 | 10.11.2021 |          |
| 5  | Проходження переддипломної практики,<br>збір та аналіз матеріалів до МКР   | 29.11.2021 | 18.12.2021 |          |
| 6  | Аналіз предметної області та розробка<br>технічного завдання. Моделювання<br>результатів   | 23.12.2021 | 02.01.2022 |          |
| 7  | Опис фахової частини МКР, зокрема<br>дослідження публікацій щодо<br>діагностування COVID-19, огляд існуючих<br>архітектур штучних нейронних мереж для<br>вирішення поставленої задачі, реалізація<br>обробних технологій з аналізом отриманих<br>результатів | 03.01.2022 | 25.01.2022 |          |
| 8  | Розробка спеціальної частини з охорони<br>праці та методичної частини  | 26.01.2022 | 30.01.2022 |          |
| 9  | Попередній захист МКР на засіданні комісії<br>кафедри  | 31.01.2022 | 31.01.2022 |          |
| 10 | Корегування роботи за результатами<br>попереднього захисту   | 01.02.2022 | 03.02.2022 |          |
| 11 | Остаточне оформлення пояснювальної<br>записки та слайдів доповіді для захисту  | 04.02.2022 | 08.02.2022 |          |
| 12 | Подання МКР рецензенту   | 11.02.2022 | 12.02.2022 |          |
| 13 | Рецензування МКР   | 13.02.2022 | 14.02.2022 |          |
| 14 | Подання МКР, її електронної копії та інших<br>документів (відгуку, рецензії) до захисту  | 16.02.2022 | 17.02.2022 |          |
| 15 | Захист МКР перед екзаменаційною<br>комісією (ЕК)   | 23.02.2022 | 24.02.2022 |          |

Розробив студент \_\_\_\_\_  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

Керівник роботи докт. техн. наук, проф. Ю. П. Кондратенко  
(наук. ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

«23» жовтня 2021 р.

# АНОТАЦІЯ

до магістерської кваліфікаційної роботи  
студента групи 607 ЧНУ ім. Петра Могили

**Савчук Олексій Андрійович**

на тему: «**ДІАГНОСТУВАННЯ COVID-19 НА ОСНОВІ МЕТОДІВ  
МАШИННОГО НАВЧАННЯ**»

Актуальність магістерської кваліфікаційної роботи обумовлена пандемією COVID-19. **Метою роботи** є діагностування COVID-19 за наявними знімками КТ на основі методів машинного навчання. **Об'єкт дослідження** – це процес медичної діагностики хвороб з використанням штучного інтелекту і методів машинного навчання. **Предмет дослідження** – згорткові нейронні мережі для розпізнавання образів на рентгенівських та КТ-знімках.

Робота складається з п'яти розділів. У першому розділі проводиться аналіз предметної сфери, а саме аналіз актуальності проблеми. Проведено аналіз наявних аналогів та публікацій та визначено основні області застосування нейронних мереж для діагностування захворювань, поставлена задача дипломної роботи. У другому розділі досліджено та описано архітектуру нейронної мережі для вирішення поставленої задачі, розібрані принципи роботи згорткових нейронних мереж. У третьому розділі був проведений вибір програмних засобів та описаний процес створення нейронної мережі. Проведено модифікацію алгоритму U-Net з використанням алгоритмів покращення зображення. У четвертому розділі проведено опис методичної частини. У п'ятому розділі описано норми до умов праці під час пандемії COVID-19. Правила поведінки під час пандемії.

Загальний обсяг роботи 101 сторінка. Магістерська робота містить 14 рисунків, 3 таблицю і посилання на 64 джерел.

*Ключові слова: комп'ютерна томографія, рентгенографія, нейронні мережі, згорткові нейронні мережі, U-Net, COVID.*

## ABSTRACT

to the master's qualification work by the student of the group 607  
of Petro Mohyla Black Sea National University

**Savchuk Oleksii**

### **“DIAGNOSIS OF COVID-19 USING MACHINE LEARNING METHODS”**

The relevance of the master's qualification work is due to the COVID-19 pandemic. **The aim of the work** is to diagnose COVID-19 on the basis of available CT-rays based on machine learning methods. **The object of research** is the process of medical diagnosis of diseases using artificial intelligence and machine learning methods. **The subject of research** - convolutional neurons of the network for pattern recognition on X-rays and CT scans.

The work consists of five sections. The first section analyzes the subject area, namely the analysis of the relevance of the problem. The analysis of available analogues and publications is carried out and the main areas of application of neural networks for diagnosing diseases are determined, the task of the diploma work is set. The second section investigates and describes the architecture of the neural network to solve the problem, analyzes the principles of convolutional neural networks. In the third section, a selection of software was made and the process of creating a neural network was described. The U-Net algorithm has been modified using image enhancement algorithms. The fourth section describes the methodological part. Section 5 describes standards for working conditions during the COVID-19 pandemic. Rules of conduct during a pandemic.

Total volume of work 101 pages. The master's thesis contains 16 figures, 3 tables and links to 64 sources.

*Keywords:* computed tomography, radiography, neural networks, convolutional neural networks, U-Net, COVID.

## ЗМІСТ

|  |    |
|--|----|
| ВСТУП.....   | 5  |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ .....   | 8  |
| 1.1 Актуальність проблеми .....  | 8  |
| 1.2 Огляд та аналіз наявних аналогів та публікацій .....                                       | 10 |
| 1.3 Постановка задачі .....  | 15 |
| Висновки до розділу 1 .....  | 17 |
| 2 МАТЕМАТИЧНІ МОДЕЛІ, МЕТОДИ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ<br>ВИРШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ ..... | 18 |
| 2.1 Архітектура згорткової нейронної мережі .....  | 18 |
| 2.2 Операція згортки.....  | 21 |
| 2.3 Агрегувальні шари.....   | 27 |
| 2.4 Повнозв'язні рівні.....  | 29 |
| 2.6 Рівень витрат та функції витрат .....  | 30 |
| 2.7 Активаційні функції .....  | 32 |
| Висновки до розділу 2 .....  | 35 |
| 3 МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ .....                                       | 36 |
| 3.1 Вибір програмних засобів .....   | 36 |
| 3.2 Підготовка набору даних для навчання.....  | 40 |
| 3.3 Модифікація алгоритму .....  | 44 |
| 3.4 Аналіз результатів.....  | 46 |
| Висновки до розділу 3 .....  | 54 |
| 4 МЕТОДИЧНА ЧАСТИНА.....   | 56 |
| 5 ОХОРОНА ПРАЦІ.....   | 75 |
| ВИСНОВКИ .....   | 93 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....  | 95 |

## **ПЕРЕЛІК СКОРОЧЕНЬ**

ПЗ - програмне забезпечення

НМ - нейронна мережа

КТ - комп'ютерна томографія

ІС - інтелектуальна система

ІІС - інтелектуальна інформаційна система

ШІ - штучний інтелект

ШНМ – штучні нейронні мережі

ЗНМ – згорткові нейронні мережі



# Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

## «ДІАГНОСТУВАННЯ COVID-19 НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ»

Спеціальність 124 «Системний аналіз»

**124 – МКР – 607.21610120**

Студент \_\_\_\_\_ Савчук О. А.  
«\_\_» \_\_\_\_\_ 2022 р.

Консультант \_\_\_\_\_ Ю. П. Кондратенко  
докт. техн. наук, проф.  
«\_\_» \_\_\_\_\_ 2022 р.

Миколаїв – 2022

## ВСТУП

Другий рік суспільство веде боротьбу із коронавірусом, який заклав на карантин більшу частину світу. Пандемія внесла свої корективи майже у всі сфери суспільства: в економіку, розваги, виробництво. Не залишилась осторонь і технологічна галузь. Велика кількість компаній перейшла на віддалену працю, але це не є рішенням для подолання проблеми.

Медпрацівникам, які стоять на варті боротьби к COVID-19, необхідно перевіряти та аналізувати сотні тисяч або навіть мільйонів медичних даних та діагнозів згідно власних знань, на кожний окремий випадок витрачається велика кількість часу за який пацієнта можна вже не врятувати. З стрімким розвитком алгоритмів машинного навчання, а саме нейронних мереж, проблему аналізу медичних даних, які є частиною медичної історії пацієнта стало можливо вирішити з високою точністю.

Медичні експертні системи дозволяють лікарю не тільки перевіряти власні діагностичні припущення, але і дають можливість звертатися до системи за консультацією в складних клінічних випадках, де необхідний аналіз великого обсягу симптомів пацієнтів.

Основними проблемами при прийнятті медичних рішень є недостатність знань, обмеженість часових ресурсів, відсутність можливості залучення великого числа компетентних експертів і неповнота інформації про стан пацієнта. Система підтримки прийняття рішення, в свою чергу, може мати в наборі інструменти для обробки даних, різні принципи побудови моделей, алгоритми розпізнавання образів, кластеризації, виділення значущих параметрів, встановлення взаємних залежностей.

На даний час для розшифрування рентгенівських зображень та комп'ютерної томографії в закладах медицини є лікар-рентгенолог, спеціаліст з діагностики променевих знімків. Висококваліфіковані фахівці-рентгенологи отримують великі гроші за свою роботу, і на їх фах колосальний попит в закладах медицини. В реаліях сьогодення ці фахівці працюють більше норм, бо попри великий попит

висококваліфікованих лікарів за цим фахом дуже мало, кожен день вони обробляють неймовірно великі обсяги даних. Такий фахівець повинен видавати діагнози дуже швидко, майже моментально, а на практиці розшифрування даних займає від кількох хвилин до кількох днів в залежності від складності випадку. У зв'язку з пандемією, медичні установи зараз перевантажені по всій земній кулі, медичного персоналу та обладнання недостатньо для швидкої протидії захворюванню.

Машинне навчання та глибоке навчання можуть значно полегшити цю складну задачу та кардинально змінити ситуацію. Штучний інтелект (ШІ) і машинне навчання вже зараз успішно застосовуються в медичній сфері, поступово перетворюючись з допоміжного інструменту для медичного персоналу в надійний, вирішують різноманітні завдання.

У сучасній ситуації задача діагностики захворювань за медичними знімками методами машинного навчання є як ніколи актуальною, що і вирішується у даній роботі. Система, розроблена в ході роботи, допоможе розвантажити медичних працівників, підвищити швидкість, точність і зручність діагностування COVID-19 за знімками рентгенографії та комп'ютерної томографії, суттєво скоротять витрати на спеціалістів з розшифрування знімків.

**Метою роботи** є діагностування COVID-19 за наявними знімками КТ на основі методів машинного навчання.

**Об'єктом дослідження** виступає процес медичної діагностики хвороб з використанням штучного інтелекту і методів машинного навчання.

**Предметом дослідження** є згорткові нейронні мережі для розпізнавання образів на рентгенівських та (або) знімках КТ.

Для досягнення поставленої мети потрібно виконати такі завдання:

- дослідити предметну область в сфері розпізнавання образів на рентгенівських та (або) знімках КТ;
- провести аналіз наявних методів та інструментів для вирішення поставленої мети;

- розробити архітектуру мережі та навчити її розпізнавати цифрові рентгенівські знімки та знімки КТ, діагностувати легеневі хвороби;
- розробити програмний застосунок на базі навченої моделі та провести тестування на запропонованих наборах, проаналізувати отримані результати.

В ході цієї роботи буде спроектована та розроблена система медичної діагностики COVID-19. Ця система буде розроблена за допомогою алгоритмів машинного навчання та класифікації, з ціллю утилізувати зібрані дані для підвищення точності визначення позитивного результату на COVID-19.

Виявлення коронавірусу на ранніх стадіях допоможе зупинити поширення пандемії. На основі аналізу рентгенограм (рентгенівських знімків та (або) знімків КТ) грудної клітки інфікованих COVID-19 пацієнтів було встановлено, що на ранніх стадіях захворювання спостерігаються слабкі симптоми, котрі супроводжуються невеликим матовим помутнінням легень та формуванням вузлів. Тому опрацювання цих медичних зображень для виявлення випадків інфікування в умовах пандемії, може дати позитивні результати у протидії коронавірусу. Аналіз рентгенограм є дуже актуальним напрямком сучасних досліджень.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ

Зараз нейронні мережі відіграють дуже важливу роль у науковому світі. Вони допомагають передбачати різні фінансові часові ряди, визначати властивості хімічних моделей за їх структурою. Методи машинного навчання активно застосовуються і у сфері медицини. Вони можуть прогнозувати схильність до різних психічних захворювань, вірогідність різних хвороб та навіть прогнозувати тривалість життя людини.

Для успішного лікування будь якої хвороби важливим аспектом є її своєчасне та правильне діагностування. Системи розпізнавання хвороб, що побудовані методами машинного навчання, можуть давати навіть більш точні результати, ніж досвідчені лікарі.

## 1.1 Актуальність проблеми

На даний час за даними «Національного центру біотехнологічної інформації» (NCBI), існує 219 видів вірусів, що можуть інфікувати людей. Коронавіруси – це група вірусів, які здебільшого спричиняють незначні проблеми, що супроводжуються симптомами кашлю та застуди. Більшість коронавірусів є нешкідливими для людей. Проте новий коронавірус COVID-19 є доволі агресивним та його похідні. Він доволі часто спричиняє смерть пацієнтів або загострення інших захворювань, що приводить до смерті.

Вірус називають короною завдяки короноподібним білковим структурам на його поверхні. Коронавіруси є одноланцюговими РНК-вірусами. Вони сильніше мутують у порівнянні з вірусами на ДНК-основі. COVID-19 поширюється швидше, ніж інші вірусні ГРВІ. При кожному контакті COVID-19 з людським організмом формується дуже міцний контакт з мембраною людської клітини завдяки білковим шипам. Інкубаційний період коронавірусу проходить впродовж приблизно 5 днів.

При COVID-19 система дихання не в змозі передавати достатньо кисню в кров людини, бо легені наповнюються рідиною, що знижує здатність організму засвоювати кисень та виводити вуглець з організму. COVID-19 визнають дуже

небезпечним тому, що при ураженні легень COVID-19 людина може не мати симптомів довгий час, мати відмінне самопочуття, але зміни у дихальних мережах пацієнта у цей час можуть бути вже не виправними. Тому актуальним є питання швидкого та якісного виявлення цього захворювання.

Можливість виявлення коронавірусу на ранніх стадіях допоможе зупинити поширення пандемії. У країнах, що розвиваються, медичні послуги важкодоступні та мають значну фінансову вартість, тому присутні обмеження, пов'язані з проведенням діагностичних та тестових процедур. На даний час використовується великий перелік аналітичних методів та засобів, котрі можуть бути ефективно застосовані в умовах пандемії.

Для визначення захворювання COVID-19 роблять одну з таких медичних процедур: рентген легень, КТ легень, УЗД грудної клітки, біопсія легені або МРТ грудної клітки. В даний час КТ грудної клітки є одною з найефективніших процедур виявлення пневмонії. Рентгенологічне зображення іноді є кращим, ніж КТ-знімки, оскільки зазвичай КТ-дослідження займає більше часу, ніж рентгенологічне, але процедура КТ є найбільш точною. В Україні для запису рентгенівського зображення використовують фіксацію його на рентген-чутливій плівці з подальшим проявом та переводом її в цифровий формат. Діагностика коронавірусу базується на даних рентгенографії грудної клітки або КТ (комп'ютерної томографії). Комп'ютерна томографія є найбільш ефективним способом діагностики COVID-19, вона також є найбільш складною та вимагає дуже технологічного обладнання.

Рентгенографія – це дослідження внутрішньої структури за допомогою рентгенівських променів на плівку або папір. Найбільш часто використовується для медичного дослідження для отримання проекційного відображення анатомічних структур організму під час проходження через них променів рентгену і фіксації ступеня проходження рентгенівського випромінювання.

Застосування рентгену – це порівняно недорогий метод діагностики. Одна з переваг рентгенівського обладнання – легкість в управлінні. Мобільність апаратів

дає можливість обстежувати пацієнта безпосередньо у палаті медичного закладу. При цьому персонал радіографічного відділення не піддається зайвій загрозі інфікування. Рентгенографія є дуже поширеним методом виявлення коронавірусу у всьому світі, особливо у розвиваючись країнах. Сьогодні також є системи, що виконують реєстрацію даних у цифровому вигляді за замовчуванням. У більшості країн світу даний спосіб є стандартом.

Для деяких задач все ще не змогли побудувати нейронні мережі, що були давали результати краще за лікарів. На одній із медичних конференцій провели експеримент, де лікарі змагалися з нейромережою у діагностиці захворювань. За результатами експерименту машина майже однаково з людиною може обробляти комп'ютерну томографію, краще за лікарів аналізує рентгенівськими знімки.

Але нейронна мережа не здатна бачити деякі речі, які б легко помітила людина. Наприклад, якщо на рентгеновому знімку лікар може помітити дефект та попросити зробити ще один, то мережа не зможе зробити такий висновок, а просто проаналізує знімок, і в наслідок може дати невірну відповідь.

Зараз штучний інтелект не може повністю замінити лікарів. Але вже зараз вони можуть допомагати лікарям приймати рішення. Агентство Frost & Sullivan відмічає, що технології допомагають збільшити точність постановки діагнозів на 30–40% та зменшити вартість медобслуговування наполовину.

## **1.2 Огляд та аналіз наявних аналогів та публікацій**

В науковій літературі подано опис обширного переліку досліджень на основі систем штучного інтелекту для різноманітних медичних цілей. Зокрема діагностування хвороби Альцгеймера, оцінка стадії перебігу ракових захворювань, аналітичне опрацювання результатів біопсії та дерматоскопії тощо. Впродовж останнього періоду часу пандемія COVID-19 спричинила значне навантаження в роботі медичних працівників. Тому формування медичних діагностичних та лікувальних систем на основі штучного інтелекту є актуальним напрямком досліджень.

Для опрацювання медичних зображень використовуються інформаційні системи сформовані на основі методів аналізу даних, зокрема методів машинного навчання та глибокого навчання. Модель глибокого навчання використовує підходи фільтрації та класифікації, подібно до людського мозку. Глибоке навчання – це підклас машинного навчання, який сформовано на основі декількох рівнів обробки даних. Окремі шари використовуються для аналізу вхідних даних та їх класифікації. Введення даних може відбуватись текстовому, звуковому або графічному вигляді. Тому глибоке навчання може ефективно застосовуватись для задач класифікації медичних зображень.

Згортка нейронна мережа (CNN) найкраще підходить для класифікації зображень при глибокому навчанні. Але на даний час існують певні концептуальні обмеження. Зокрема, під час роботи максимального пулу CNN втрачається інформація про позицію об'єкта, яка використовується мережею для розпізнавання. Крім того, CNN не розглядає кілька просторових зв'язків між простішим об'єктом. Щоб подолати ці концептуальні обмеження CNN, автори використовували згорткову капсульну мережу (CapsNet) для класифікації рентгенівських зображень.

У роботі автори запропонували модель глибокого навчання на основі групової капсульною мережею візуальної геометрії (VGG-CapsNet) шляхом поєднання VGGNet і CapsNet для видобування детальнішої інформації з рентгенівських зображень.

В роботі з назвою «Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning» була наведена модель глибокого навчання для обробки рентгенівських зображень з ціллю діагностування пневмонії та інших легеневих захворювань. Ця модель має точність тестування майже у 92%.

В схожому дослідженні був проілюстрований досить якісний підхід для здійснення класифікації COVID за допомогою методів глибокого навчання на основі моделей побудованих на згорткових нейронних мережах, структура такої мережі нараховувала чотири згорткових шара разом з двома повнозв'язними. Разом



з використанням додаткових методів покращення навчання, таких як механізм доповнення даних для навчання, досягла значення у 93.7% точності.

Згорткова нейромережа створена групою авторів з Корнельського університету, що використовується для автоматичного діагностування COVID-19 на основі попередньо навчених моделей, що базуються на аналізі рентгенівських зображень. В рамках цього дослідження група використовувала попередньо навчені алгоритми ResNet50, InceptionV3 та Inception-ResNetV2 і техніку transfer learning, щоб отримати найкращу точність діагностування на аналізі знімків і досягли 96% точності.

Таким чином можна зробити висновок, що сучасні глибокі згорткові нейроні мережі без особливих модифікацій досягають більш ніж 90% точності у задачі діагностування захворювань легень шляхом аналізу рентгенівських знімків. Для порівняння, згідно дослідження Лондонського коледжу радіології, в ході якого прийняли участь майже 140 кваліфікованих лікарів, відсоток вірно діагнозів респіраторних захворювань за рентгенівськими знімками групою висококваліфікованих лікарів склав близько 82.7%.

У роботі «Apostolopoulos I.D., Mpesiana TA. COVID-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks.» запропонована модель виявлення COVID-19 за допомогою аналізу рентгенографічних зображень грудної клітки. Модель розроблена на основі глибокого навчання з CNN, зокрема, передача процедури навчання. Для навчання моделі автором використано рентгенівські зображення грудної клітки здорової людини, інфікованої пневмонією людини та інфікованих COVID-19 пацієнтів. Зазначені зображення загальнодоступні в медичних сховищах даних. Автори розробили та протестували модель за допомогою двох наборів рентгенівських зображень. Кожен з використаних наборів містить рентгенограми понад двохсот двадцяти інфікованих COVID-19 пацієнтів, понад сімсот пацієнтів з пневмонією та понад п'ятсот здорових громадян. Після аналізу результатів моделювання автори публікації виявили, що рівень точності моделі досягає 96,78% при пошуку

інфікованих COVID-19 пацієнтів. В перспективі автори планують включити до процесу навчання колекції та набори рентгенівських знімків пацієнтів інфікованих ГРВІ, ЕБОЛА тощо.

Холл в «Hall L.O., Paul R., Goldgof D.B., Goldgof G.M. Finding COVID-19 from chest X-Rays using deep learning on a small dataset.» обговорює важливість аналітичного опрацювання рентгенографії грудної клітки для процесів раннього діагностування та виявлення COVID-19. Автором запропоновано модель глибокого навчання, котру сформовано на основі комбінації попередньо підготованих мереж на основі «VGG16», «Resnet50» та CNN. В процесах навчання та тестування використано загальнодоступні колекції рентгенограм грудної клітки понад тридцяти інфікованих COVID-19 пацієнтів і понад трьохсот інфікованих звичайними ГРВІ та пневмонією пацієнтів. Автори використали для попереднього навчання Resnet50 понад сотню рентгенівських зображень COVID-19 пацієнтів та понад сотню рентгенівських зображень пацієнтів хворих на пневмонію. Запропонована авторами комбінація сформована на основі трьох різних класифікаторів перевірена на понад тридцяти невидимих рентгенівських знімках COVID-19 пацієнтів та понад двохсот зображеннях пацієнтів з ГРВІ та пневмонією. Задекларована авторами точність гібридної моделі становить 91,24%.

Робота Чжан «COVID-19 screening on chest X-Ray images using deep learning based anomaly detection» присвячена ефективності опрацювання рентгенограм грудної клітки для виявлення інфікованих COVID-19 пацієнтів. Оскільки цей вірус активно вражає органи дихання та зокрема легені. Автор запропонував сформовану на основі глибокого навчання модель виявлення легеневих аномалій. У дослідженні використано сто доступних на Github рентгенологічних зображень грудної клітки. В досліджуваному наборі даних сімдесят рентгенологічних зображень грудної клітки були отримані для COVID-19 пацієнтів. Решта тридцять зображень були отримані для пацієнтів, що страждають на ГРВІ, пневмонію тощо. Для ефективного використання можливостей методів на основі глибокого навчання автори використали понад тисячу зображень отриманих для хворих на пневмонію

пацієнтів. Опубліковані результати моделювання дозволили забезпечити точність ідентифікації COVID-19 на рівні 96%ю Водночас, для не COVID-19 випадків становить 70,65%.

У роботі «COVIDiagnosis-net: deep BayesSqueezeNet based diagnostic of the corona virus disease 2019 (COVID-19) from X-ray images» Акар та Коркмаз обговорюють процеси виявлення COVID-19. Автори використали сформовану на основі штучного інтелекту структуру для виявлення COVID-19 на основі аналізу рентгенограм грудної клітки. Було використано «Squeezenet» та байєсівську оптимізацію для формування моделі глибокого навчання. Для підвищення точності використано гіперпараметри та доповнені набори даних. Автори стверджують про досягнення точності 98,3% при виявленні COVID-19, пневмонії та нормальних випадків.

Халіфа у роботі «Detection of corona virus (COVID-19) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using chest X- Ray dataset» досліджує процеси виявлення спричиненого COVID-19 запалення легень. Автор описує використання генеративної змагальної мережі (GAN) в комплексі з моделями навчання глибокої передачі «AlexNet», «Resnet18», «Squeezenet» та «GoogleNet». Для навчання та динамічного тестування моделей було використано набір даних в якому понад п'ять тисяч вісімсот рентгенограм грудної клітки інфікованих пневмонією або здорових пацієнтів. На основі результатів моделювання авторами зроблено висновок, що точніть комбінації GAN з «Resnet18» досягає 99% відповідно до шкали точності, запам'ятовування та оцінки (F-1).

Салман у роботі «Detection Using Artificial Intelligence; United States: The DSpace Institutional Digital Repository System; 2020.» опублікував дослідження рентгенівських зображень високої чіткості для зменшення робочого навантаження радіологів при виявленні COVID-19. Було використано набір з двохсот шестидесяти доступних на Kaggle та Github зображень. Розроблено моделі глибокого навчання на основі CNN. Використаний набір зображень містить

рентгенограми грудної клітки сто тридцяти здорових громадян та сто тридцяти інфікованих COVID-19 пацієнтів. Автор стверджує, що завдяки результатам моделювання було досягнуто показники точності, які еквівалентні експерту-радіологу.

Магхдід «COVID-19 pneumonia from X-Ray and CT images using deep learning and transfer learning algorithms.» зосередився на формуванні загальнодоступних наборів даних рентгенограм грудної клітки. Автор розробив метод діагностування COVID-19 з використанням трансферного навчання на основі моделей глибокого навчання. Розроблено дві моделі прогнозування з використанням CNN на базі «Alexnet». В опублікованих матеріалах досягнуто точність 94,1% для моделі CNN на основі потоку та 98% для моделі CNN на базі «Alexnet».

Бассі та Аттукс «A deep convolutional neural network for COVID-19 detection using chest X-rays» запропонували модель класифікації рентгенографії грудної клітки для виявлення COVID-19. Розроблена модель на основі «DenseNet121 CNN», використовує рентгенографічні зображення грудної клітки COVID-19 пацієнтів, хворих на пневмонію та здорових громадян. Описана в роботі модель пройшла два етапи навчання. На першому етапі навчання відбувалося за допомогою набору зображень. А під час другого етапу навчання відбувалося на основі бази даних рентгенограм грудної клітки. На основі результатів симуляційного дослідження автори повідомляють про досягнення точності 97,8% для COVID-19 тестів.

### **1.3 Постановка задачі**

Перед початком проектування системи необхідно проаналізувати існуючі аналоги реалізованих в даній сфері систем і сформулювати функціонал майбутньої системи. Далі слід розбити функціонал системи на окремі компоненти та визначити план майбутніх робіт і список реалізації необхідного функціоналу в системі.

Після проведення теоретичних досліджень була сформульована концепція завдання – реалізація підсистеми прийняття рішень для ранньої діагностики захворювань.

Список необхідних кроків при вирішення завдань за допомогою штучних нейронних мереж можна сформулювати наступним чином:

- сформулювати задачу, що буде вирішуватись нейронною мережею: класифікація, прогнозування, кластеризація);
- визначити обмеження задачі (швидкість, точність відповіді);
- встановити вхідні данні (тип: зображення, звук, розмір: 100x100, 30x30, формат) і вихідні дані (кількість класів);
- з'ясувати оптимальну архітектуру нейронної мережі, потрібну для вирішення поставленої задачі.

Перед даною нейронною мережею ставиться задача – класифікувати зображення. А саме мова йде про – класифікацію зображень, отриманих за допомогою рентгенівських апаратів. Специфіка застосування обладнання потребує певних обмежень для задачі, що розв'язується. Це швидкість відповіді – до 1 секунди, а також ступінь точності розпізнавання. Припускається точність не менше, ніж 90%. У якості вхідних даних будуть виступати цифрові зображення в форматі jpeg. Їхні розміри – 150x150 пікселів. У результаті будуть отримані два класи, які визначатимуть виявлений COVID-19 на знімку, що аналізується, або негативний результат. Створена система потенційно полегшує та автоматизує роботу працівників медичних закладів у процесі діагностики захворювань за рентгенівськими зображеннями. Система виконує поставлену задачу, використовуючи методи машинного навчання. Зокрема потрібно сконцентруватись на практичному застосуванні нейронних мереж задля вирішення описаної вище задачі..

Після детального аналізу предметної області, існуючих систем та проблем задачі класифікації рентгенівських зображень був створений детальний план реалізації даного проекту. План проектування і розробки даної системи відноситься до ітеративних та включає в себе наступні етапи:

- проведення теоретичних досліджень та дослідження існуючих на сьогодні методів глибокого машинного навчання в контексті вирішення задач, пов'язаних із охороною здоров'я;
- виявлення та підготовка набору даних, що необхідні для навчання алгоритмів. Важливо також провести дослідження технік та методів попередньої обробки даних, які здатні позитивно вплинути на якість навчання алгоритмів;
- аналіз та відбір найбільш релевантних алгоритмів для класифікації зображень. Необхідно сфокусуватись на застосуванні штучних нейронних мереж для вирішення зазначеної задачі;
- проведення необхідних експериментів, реалізація на практиці експериментального моделювання та навчання моделі;
- аналіз отриманих результатів;
- проведення порівняльного аналізу роботи різних алгоритмів.

### **Висновки до розділу 1**

В даному розділі були розглянуті:

- питання актуальності проблеми рекомендацій;
- проаналізовано попередні дослідження та публікації;
- поставлена задача.

## **2 МАТЕМАТИЧНІ МОДЕЛІ, МЕТОДИ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ**

### **2.1 Архітектура згорткової нейронної мережі**

Згорткові нейронні мережі (ЗНМ, англ. convolutional neural networks, CNN) – це підкатегорія глибоких нейронних мереж, що здебільшого застосовується для розпізнавання об'єктів та класифікації зображень. Принцип роботи згорткових нейронних мереж будується на сприйманні та обробці даних у вигляді тензорів. Цей процес уможливорює роботу з даними зображень у природному стані. Кожне з вхідних зображень характеризується наявністю наступних параметрів: ширини, довжини, а також глибини, що визначається кодуванням зображення. Найбільш поширеним типом кодування зображення є RGB-кодування. Принцип роботи полягає у тому, що колір кожного пікселя кодується за допомогою значень для червоного, зеленого і синього кольорів. Такі категорії значень, що описують зображення, називають каналами. В процесі роботи мережі утворюється інший вимір тензорів. У ньому містяться карти виявлених ознак зображення. Таким чином, кожне зображення аналізується згортковими нейронними мережами як чотиривимірний масив даних. У більшості випадків для визначається конкретна ширина і довжина зображень, необхідні для кожної мережі. Однак існують і мережі, у яких може відбуватися масштабування.

У випадках, коли необхідно розпізнати великі обсяги даних в умовах використання, необхідно забезпечити високу здатність моделі до навчання, а також великий відсоток правильних припущень щодо ознак зображення. Порівняно із традиційними нейронними мережами прямого поширення, що мають схожу кількість шарів, згорткові нейронні мережі відрізняються більш високою здатністю до навчання. Це обумовлено значно меншою кількістю параметрів та зв'язків. Задля забезпечення правильного навчання в задачах розпізнавання образів традиційними повнозв'язними нейронними мережами необхідна значно більша кількість даних. Це пов'язано з тим, що усі вхідні зображення можуть мати

розмірність мінімум у декілька сотень тисяч. У зв'язку з цим необхідна наявність відповідної кількості прикладів з різними значеннями для кожного виміру.

Ще одна відмінність ЗНМ полягає у тому, що вони зазвичай не вимагають попередньої обробки даних. Ця обставина обумовлена тим, що згорткові нейронні мережі використовують дані зображень напряму. Це уможливорює спрощення структури мережі, із урахуванням упорядкованості початкових даних. Водночас, незважаючи на велику кількість сильних сторін, їх застосування все ще вимагає застосування великих обсягів ресурсів, особливо у випадках, коли необхідно розв'язати задачі із зображеннями з високою роздільною здатністю. У зв'язку з цим виникає потреба у розробці ЗНМ із високим ступенем оптимізації операцій.

Згорткові нейронні мережі застосовуються не лише для виконання задач з розпізнаванням та класифікацією зображень. Завдяки модифікаціям ЗНМ широко застосовуються для задач, пов'язаних із обробкою природної мови (ОПМ, англ. Natural language processing, NLP). Зокрема, вони демонструють високий рівень ефективності в задачах, пов'язаних із семантичним парсингом, а також у процесі моделювання та класифікації речень.

Структура ЗНМ відповідає загальноприйнятій структурі нейронної мережі. До складу мережі входять: вхідний шар, певна кількість прихованих шарів та вихідний шар. До складу прихованих шарів традиційно входять згорткові шари, шари агрегування (субдискретизації), нормалізуючі та повнозв'язні шари. Такі шари пов'язуються один з одним за допомогою шарів з визначеними активаційними функціями. Головним елементом ЗНМ є згорткові шари, де по відношенню до даних з попереднього шару виконується операція згортки. Більш детально вона розглянута в розділі Загальна структурна схема побудови згорткової нейронної мережі зображена на рис. 1.1.



Кафедра інтелектуальних інформаційних систем  
Діагностування COVID-19 на основі методів машинного навчання

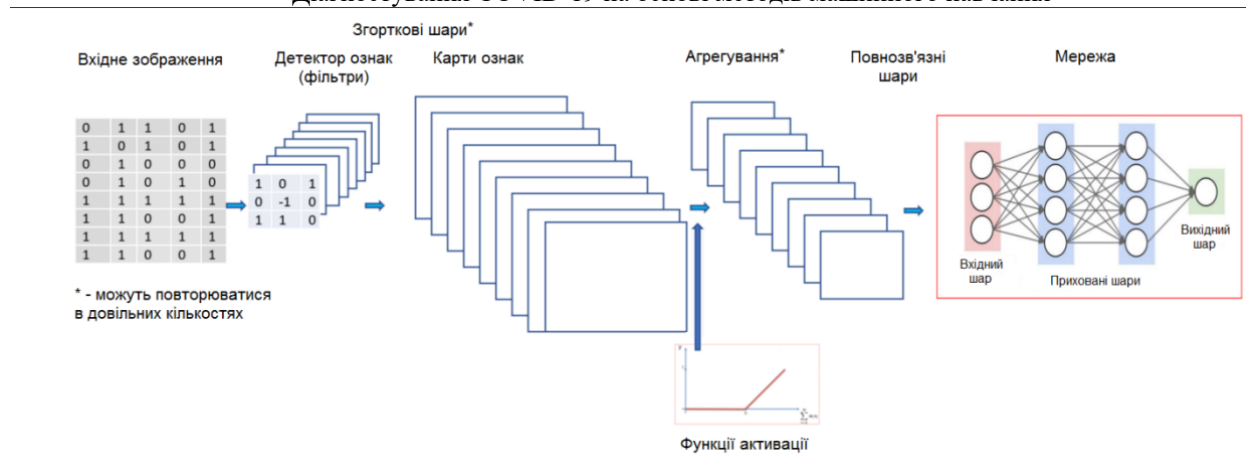


Рис. 1.1. Загальна структурна схема побудови згорткової нейронної мережі

Застосування згорткових нейронних мереж набуло високого рівня популярності у задачах розпізнавання зображень. Це зумовлено тим фактом, що ЗНМ уникають або зменшують основні проблеми, пов'язані із обробкою зображень в традиційних повноз'язних мережах. Чисельні зв'язки в таких нейронних мережах та велика кількість вхідних параметрів у кожному із зображень вимагають великої кількості вхідних прикладів. У випадку їх нестачі швидко виникає проблема у вигляді перенавчання мережі. Це означає, що модель занадто ускладнюється та навчається на нехарактеристичних другорядних ознаках, що були знайдені у вхідних даних. Ще більш серйозною проблемою є та обставина, що такі моделі не є демонструють стійкості до збурень або будь-яких змін у зображеннях. У якості прикладу можна навести розташування об'єкта в іншій частині зображення або зміну кута погляду на нього. Слід також зазначити, що подібні мережі не здатні враховувати топології вхідних параметрів: зображення є даними, де розташовані близько один до одного пікселі мають високу кореляцію. Такий рівень кореляції робить необхідним знаходження локальних ознак та взаємозв'язків для того, щоб зробити процес визначення образів ефективним.

В ЗНМ вказані вище проблеми вирішуються завдяки використанню кількох рішень. Головним з них є обробка вхідних даних на прихованих рівнях. Такі рівні розподіляється на частини. Кожна з них сприймає виключно дані попереднього рівня, що локалізуються в певній невеликій області. Області такого типу

називаються локальними рецептивними полями. За допомогою локальних рецептивних полів, мережа здатна визначати найбільш прості елементи групи пікселів та параметри зображення, як повороти або сполучення між ділянками зображення. Ще одним рішенням в основі згорткових нейронних мереж є спільні ваги. Їх використання зменшує рівень чутливості мережі до змін у поточному положенні об'єктів на зображенні, незначних поворотів його спотворення. Таким чином, детектор певної простої ознаки зображення, що застосовується на конкретній його ділянці, здатен визначати аналогічні ознаки і на іншій ділянці. Завдяки цій особливості ваги детекторів для локальних рецептивних полів на різних ділянках будуть однаковими. Наслідком цього в різних частинах зображення для схожих ознак відбуваються однакові зміни. Слід зауважити, що важливим елементом є операція субдискретизації. Вона дає можливість зменшувати загальну розмірність даних. При цьому стає меншою рівень їх чутливості до суттєвих збурень. Частіше за все субдискретизація чергується із операціями згортки в архітектурі мережі. Завдяки цій обставині поступово зменшується розмірність даних та збільшується кількість карт ознак.

## 2.2 Операція згортки

Операції згортки – це найголовніші елементи згорткових шарів нейронної мережі. Згортка – це математична операція, що за допомогою ядра згортки (детектор ознак, англ. kernel або feature detector)  $k$  на вхідні параметри у виді зображення (у вигляді тензора  $x$ ) отримує у якості результату карту ознак  $m$  (англ. feature map). Кожний із детекторів ознак складається з певної встановленої кількості фільтрів (англ. filters). Кожен з цих фільтрів в свою чергу виступає у ролі тензора параметрів з розмірністю, що зазвичай дає можливість покривати лише невелику ділянку зображення в ширину та висоту. При цьому обов'язкова відповідність розмірності у глибину. Такі фільтри і є параметричною частиною функціональності нейронної мережі, яка навчається. Фільтри також можуть бути використані не лише задля виявлення ознак та переходів. Також вони можуть виступати у ролі інструментів для збільшення різкості чи розмивання зображення.

Слід зауважити, що саме число фільтрів також може виступати у ролі одного із гіперпараметрів нейронної мережі. Це пов'язано з тим, що збільшення кількості фільтрів призводить до збільшення кількості потенційно виявлених ознак та закономірностей. При цьому завелика їх кількість потенційно призводить до повторень та знаходження ознак, що не є характеристичними. Іншими словами, наслідком може бути ускладнення та потенційного перенавчання моделі. Це означає, що кількість фільтрів відповідає кількості карт ознак, яка визначає глибину вихідних даних, а, разом з тим, і глибину вхідних для наступного рівня мережі. Це відповідно визначає кількість необхідних параметрів і на наступній стадії.

У процесі навчання фільтри отримують можливість виявляти ознаки зображення, що, групуючись, утворюють карти ознак нейронної мережі. Один фільтр відповідає за конструювання однієї карти ознак. Процес формування карти ознак відбувається завдяки переміщенню фільтра зображенням та визначення ознаки для сусідніх рецептивних полів. Крок переміщення  $s$  (англ. stride) може бути різним і є одним з параметрів нейронної моделі. Можна зробити висновок, що абсолютне положення кожної ознаки на зображення на карті не зберігається, проте ця обставина не має значення. Це пов'язано з тим, що відносне положення ознак одна відносно одної залишається незмінним. Відповідно, властивості зображення залишаються сталими.

В загальному вигляді операція може бути записана математично таким чином:

$$m(t) = (x * k)(t) = \sum_{a=-\infty}^{a=+\infty} x(a)k(a + t) \quad (2.1)$$

Значення кожного детектора ознак за межами його визначеної області означення приймаються за базові (за нульові). Це означає, що нескінченна сума може бути замінена на скінченну зі значеннями в заданій області. Використовуємо властивість комутативності для операції математичної згортки і в результаті

отримуємо спрощену функцію. Ця функція придатна для того, щоб використовувати її в нейронних мережах та створювати детектори для згорткового шару. Вона є спорідненою із взаємкореляційною функцією для значень зображення:

$$M(i, j) = (K * X)(i, j) = \sum_m \sum_n K(m, n)X(i - m, j - n) \quad (2.2)$$

де  $M(i, j)$  – елемент карти ознак з координатами  $i$  та  $j$ ,

$X$  – вхідне зображення,

$K$  – детектор ознак,

$m, n$  – розмірності детектора ознак.

Фільтру операції згортки для двовимірного випадку:

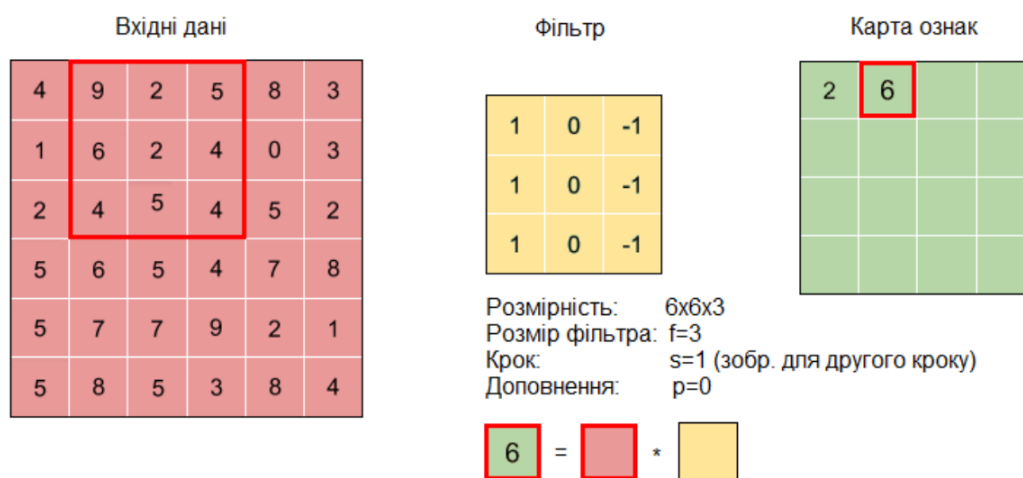


Рис. 2.1. Одновимірного фільтру для операції згортки для зображення 6x6 з трьома каналами, розміру фільтра 3, розміром кроку 1 та відсутності доповнення нулями

Звідси можна зробити висновок, що кожний з елементів карти ознак може бути інтерпретований як результат перетворення з використання скалярного добутку значень деякої області (рецептивного поля нейрона), що пов'язаний з сусідніми значеннями з тієї ж карти.

Ознаки, здатні до виявлення фільтрів, відрізняються в залежності від складності мережі, на якому вони базуються. На перших рівнях можуть мати місце прості переходи кольору або зміни освітленості. При цьому на більш глибоких рівнях фільтрами можуть виявлятися складні кути, закруглення чи образи. Такі процеси стають можливими завдяки застосуванню на наступних рівнях результатів згорток на попередніх, а також завдяки збільшенню кількості точок початкового зображення. Такі точки безпосередньо впливають на кожне із значень карти ознак на більш глибоких рівнях. Така властивість збільшує ступінь впливу певної ознаки на результат при її більш близькому розташуванні до центра зображення. Це зумовлено тим, що значення, які відносяться до подібної ознаки, будуть мати вплив на кількість значень окремої карти ознак, ніж ознака ближче до краю зображення. Це відбувається завдяки охопленню більшої кількості рецептивних полів фільтрів.

Стадія обходу зображення (англ. *stride*)  $s$  – це один з гіперпараметрів мережі. Це означає, що він є параметром, який має бути визначений ще у процесі навчання, під час конструювання моделі. Наслідком збільшення кроку є зменшення розмірності вихідних даних (тобто, до зменшення рівня складності моделі), а також до зменшення накладання рецептивних полів для різних значень.

Процес доповнення нулями (англ. *padding*)  $p$  також належить до гіперпараметрів і може бути застосованим у ситуаціях, коли обставини роблять небажаним зменшення розмірності карт ознак у порівнянні з вхідними даними, а також для її загального контролю. Принцип роботи полягає у розширенні вхідного зображення за допомогою визначеної кількості нульових значень з кожного краю. Використання доповнення нулями дає можливість збільшувати вплив значень зображення, що розташовуються з його краю. Це пов'язано з тим, що без його використання крайні значення будуть застосовуватися в картах ознак лише один раз. Відповідно, втрачається інформація з країв зображення.

Вхідні дані

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 4 | 9 | 2 | 5 | 8 | 3 | 0 |
| 0 | 1 | 6 | 2 | 4 | 0 | 3 | 0 |
| 0 | 2 | 4 | 5 | 4 | 5 | 2 | 0 |
| 0 | 5 | 6 | 5 | 4 | 7 | 8 | 0 |
| 0 | 5 | 7 | 7 | 9 | 2 | 1 | 0 |
| 0 | 5 | 8 | 5 | 3 | 8 | 4 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис. 2.2. Приклад застосування доповнення нулями на 1 шар

Процес визначення розмірності фільтрів проводиться із з розрахунком, що зі фільтри знаходять більші ознаки, що частіше за все зустрічаються рідше, але 21 можуть бути більш вагомими. В сучасних дослідженнях просліджується тенденція до її зменшення. Це пояснюється тим, що такий підхід дає можливість відповідно зменшити використання ресурсів, оскільки кількість параметрів у фільтрах, що відрізняються збільшеною розмірністю зростає квадратично. З метою збереження розмірів рецептивних полів у подібних випадках застосовується вертикальне послідовне розташування згорток. Цей процес робить можливим покращення роботи мережі. Ця обставина пояснюється тим, що у такий спосіб зростає ступінь її глибини, замість ширини для згорток із великою розмірністю. У деяких випадках застосовуються навіть згортки з розмірністю 1 задля того, щоб зменшити глибину даних перед операцією з великими витратами користувацьких ресурсів.

Розмір карт ознак знаходиться в залежності від параметрів та розраховується наступним чином:

$$d_m = \frac{d_x - f + 2 * p}{s + 1} \quad (2.3)$$

де  $d_m$  – розмірність карт ознак (ширина або висота),

$d_x$  – відповідна розмірність вхідних даних,

$f$  – розмірність фільтрів,

$p$  – доповнення нулями,

$s$  – розмір кроку обходу зображення.

Крім того, можуть паралельно застосовуватися фільтри з різною розмірністю на одному рівні. Такий тип згорток називається груповими згортками (англ. *grouped convolution*). При його застосуванні обов'язково групувати вихідні дані для кожної розмірності окремо від інших. Цей процес вимагає також використання згорток до цих груп на наступних рівнях. Такий підхід збільшує рівень ефективності мережі. Причина полягає у тому, що кожна із груп згорток виконує функцію окремого згорткового шару та відрізняється широким застосуванням у сучасних архітектурах.

Також у деяких випадках здійснюється операція розширення (англ. *dilation*). Її принцип полягає у застосуванні елементів фільтру по відношенню не до сусідніх елементів зображення, а до елементів, що мають визначений крок. Цей процес уможливорює значне зменшення розмірності даних, якщо вони містять багато надлишкової інформації. Також стає можливим узагальнення ознак.

Після застосування фільтрів та проведення усіх необхідних операцій серед описаних вище, до отриманих даних додається зсув  $b$ , що також є одним із параметрів мережі. Всі виконані зміни є лінійними. У зв'язку з цим для створення нелінійних змін у даних застосовується функція активації.

Загальний процес перетворень у згортковому шарі можна описати за допомогою рівняння:

$$y = f(w * x + b) \quad (2.4)$$

де  $y$  – вихідний результат обчислень на згортковому шарі,

$w$  – детектор ознак,

$x$  – вихідні дані,

$b$  – зсув.

### 2.3 Агрегувальні шари

Агрегувальні шари або шари субдискретизації (англ. pooling layers) – це один із основних структурних елементів згорткових нейронних мереж разом із згортковими шарами. Такі шари можуть бути як глобальними, так і локальними. Це означає, що вони можуть розповсюджуватись лише на окремі групи вхідних даних, що будуть виконувати функції рецептивних полів для нейронів поточного шару. Головна задача агрегувальних шарів – це зменшення розмірності даних із одночасним збереженням найважливіших характеристик. Це досягається завдяки формуванню відповідностей між кількома елементами (нейронами) з попередніх шарів з одним елементом даного шару. У зв'язку з цим при побудові мережі вони зазвичай застосовуються з певною періодичністю між згортковими шарами. Слід мати на увазі, що агрегувальні шари не лише зберігають глибину вхідних даних, але при цьому значно запобігають процесу перенавчання.

Існує два підтипи агрегувальних шарів: усереднювальні (англ. average) та максимізаційні (англ. maximal). У деяких випадках спостерігається застосування і мінімізаційних шарів (англ. minimal) або агрегувальних шарів за L2-нормою.

Завдяки усереднювальним шарам кожен елемент набуває середнього значення серед тих, що належать до відповідної йому групи нейронів з попереднього шару. Максимізаційні шари у свою чергу забезпечують максимальне значення. Формально операції для обох типів агрегування записуються так:

для усередненого агрегування,

$$p(i, j) = \frac{\sum_{m, n} x(i - m, j - n)}{m * n} \quad (2.5)$$

для максимізаційного агрегування,

$$p(i, j) = \max_{i, j} (x(i - m, j - n)), \quad (2.6)$$



де  $p(i,j)$  – значення елемента поточного рівня з координатами  $i$  та  $j$ ,

$x$  – вхідні дані з попередніх рівнів,

$m, n$  – розмірність рецептивного поля.

Одна з важливих функцій агрегувальних шарів – це збільшення ступеню стійкості мережі до збурень у даних. У якості прикладу можна навести зміну кута погляду на об'єкт або зміни його положення. Вибір значення з множини за визначеним критерієм, який ми знаємо, (усереднювальним або максимізаційним), забезпечує умови, за яких зміна вхідних даних на невеликі значення має незначний вплив на результати роботи агрегувального рівня. Так, схожі ознаки, матимуть аналогічні вихідні значення. До таких ознак можна віднести, наприклад, переходи, кути або заокруглення. Цей процес відбувається незважаючи на деякі зміни в положенні або освітленні. Кращі результати у подібних ситуаціях показує агрегування максимізації.

Агрегувальні шари мають декілька однакових гіперпараметрів з операцією згортки. Так, необхідно визначити крок обходу зображення, що у багатьох випадках обирається таким чином, щоб можна було уникнути процесу накладання; доповнення нулями, а також розмірність рецептивного поля, не зважаючи на той факт, що в переважній більшості мереж застосовується вікно із розмірами  $2 \times 2$ .

Одна з особливостей агрегувальних шарів полягає в тому, що вони можуть суттєво зменшувати розмірності вхідних даних та рівень впливу окремих елементів з невеликими значеннями на результат. Можна зробити висновок, що застосування такого малого рецептивного поля є цілком виправданим. Існують також архітектури мереж, у яких не використовуються агрегувальні шари. Вони замінюються згортковими шарами, відбувається збільшення кроку.

Розмірність результуючих даних для агрегувальних шарів може бути вирахована за допомогою значень параметрів аналогічно до обчислення розмірнів карт ознак для рівнів згортки.

У деяких ситуаціях замість операції агрегування застосовується процес субдискретизації (англ. *subsampling*). Його суть полягає в сумуванні елементів для

рецептивного поля кожного з нейронів. Після обчислень ця сума множиться на параметр з наступним додаванням параметра зсуву та застосуванням нелінійної функції активації. Така операція може бути записана наступним чином:

$$p(i, j) = f(w * \sum_{m,n} x(i - m, i - n) + b), \quad (2.7)$$

де  $p(i, j)$  – значення елемента поточного рівня з координатами  $i$  та  $j$ ,  
 $x$  – вхідні дані з попередніх рівнів,  
 $m, n$  – розмірність рецептивного поля.

#### 2.4 Повнозв'язні рівні

Повнозв'язні рівнями (англ. fully-connected layers) у згорткових нейронних мережах – це рівні, у яких усі нейрони з наступного шару поєднуються за допомогою зв'язків із нейронами попереднього шару. Це відбувається аналогічно до більшості шарів у звичайних нейронних мережах. Застосування даних рівнів на початкових та прихованих рівнях мережі не є ефективним, оскільки воно призводить до ускладнення моделі і навіть може ігнорувати ознаки та закономірності, що були знайдені раніше із застосуванням згорток та агрегації. Однак повнозв'язні рівні частіше за все застосовуються на передостанньому кроці роботи нейронної мережі для знаходження результату на виході мережі.

З метою забезпечення роботи повнозв'язного рівня виконується випрямління елементів матриць та шарів з попереднього кроку у послідовність значень. Такі шари, аналогічно до згорткових, розраховують скалярний добуток даних та параметрів з додаванням параметру зсуву. Саме тому загальна формула повнозв'язного рівня має схожі риси із формулою згорткового:

$$y = f(w^t * x + b) \quad (2.8)$$

Відмінність полягає у тому, що  $w$  означає не детектор ознак, а масив параметрів рівня. На повнозв'язних рівнях використовується функція активації, результат якої буде використовуватися на наступному рівні, або слугуватиме для

обчислення результатів мережі на виході. Це пояснює той факт, що типи активаційних функцій, які застосовуються на даному рівні, мають багато відмінностей із тими, що використовуються на згорткових та агрегаційних. Також вони відрізняються між собою в залежності від положення рівня та загальної мети мережі. Таким чином, одними з найбільш вживаних функцій активації для формування виходів є функції сігмоїди та Softmax. Повнозв'язні рівні можуть бути представлені, як згорткові рівні з розмірністю детектора ознак  $1 \times 1$ , де реалізовані усі зв'язки.

## 2.6 Рівень витрат та функції витрат

Рівень витрат (англ. loss layer) зазвичай стає останнім рівнем мережі. Він відповідає за оновлення параметрів у процесі її тренування. На даному рівні за допомогою заздалегідь визначеної функції витрат (англ. loss function) розраховується різниця між справжніми значеннями даних (справжніми даними) та результатами роботи мережі (прогнозом). Після закінчення процесу навчання мережі при наступному етапі тренування мережі ці дані використовуються для подальшого тренування алгоритма.

Вибір та задання функції витрат – це один із найбільш важливих елементів побудови нейронної мережі, оскільки від цього значною мірою залежить швидкість та якість тренування мережі, а також здатність мережі до оцінювання точності результатів, що отримуються. Значення функції витрат – це критерій оцінки ефективності роботи мережі. Це обумовлено тією обставиною, що за її допомогою переваги, недоліки та неточності мережі мають бути приведені до одного значення. Саме це значення має характеризувати мережу так, щоб результати роботи для різних варіантів моделі можна було впорядкувати за таким параметром як точність. Слід також зауважити, що функція витрат має враховувати усі особливості конкретної задачі, для якої застосовується мережа.

Два головні типи функцій витрат для нейронних мереж – це функції перехресної ентропії (англ. cross-entropy functions) та функції середньоквадратичної похибки (англ. mean-squared error functions). Однак для

згорткових мереж в задачах класифікації з метою зіставлення результатів роботи мережі з одним з визначених класів застосовуються імовірності того, що вхідні дані належать кожному з визначених класів. У задачі такого типу знаходження параметрів відбувається за принципом максимальної правдоподібності. Це пов'язано із тим, що нейрона мережа за допомогою наближення до еталонних параметрів намагається наблизити розподіл результатів до розподілу фактичних даних. У результаті цього, шуканим критерієм помилки роботи мережі стає різниця між імовірностями належності вхідних даних до всіх належних класів, що були отримані мережею та заданих у вибірці. Саме тому для задач класифікації зі згортковими мережами застосовується переважно тільки функції перехресної ентропії. Крім того, функція сереньоквадратичного відхилення для задачі класифікації може бути різновидом функції перехресної ентропії, у разі, коли дані розподіляються за нормальним законом.

Функція перехресної ентропії, що застосовується як функція втрат для задачі класифікації, називається функцією логарифмічних втрат (англ. logloss function). Вона застосовується паралельно із застосуванням функцій активації Softmax (для багатокласової класифікації) та сігмоїди (для бінарної класифікації) на останньому рівні мережі для обчислення прогнозів.

Її можна бути знайти за формулою:

$$L(p, y) = - \sum_n y_n \ln p_n, n \in [1, N] \quad (2.7)$$

де  $p$  – отримані за допомогою мережі ймовірності належності даних до вибраних класів,

$y$  – справжні значення для вхідних даних,

$N$  – кількість класів.

## 2.7 Активаційні функції

Функції активації або активаційні функції (англ. activation functions) застосовуються у якості нелінійного фактору після згорткових, агрегаційних, а інколи й інших шарів згорткових нейронних мереж за допомогою формування їх остаточних вихідних даних. Вибір та використання функцій активації сильно впливає на роботу мережі. Це пояснюється тим, що без їх застосування мережа не могла би навчатися розпізнавати нелінійні закономірності та розподіли. Також слід зауважити, що у випадку, коли рівні нейронної мережі не мають функції активації між собою, буде створюватись залежність лінійного типу між початковим та кінцевим рівнями одного проміжку. Описаний процес фактично робить значення кількості рівнів між ними не впливовим, тобто ці рівні стають еквівалентними одному. Крім того, за допомогою правильного вибору активаційної функції можливий контроль того, який вплив на результати буде мати елемент (нейрон) в залежності від його вхідного значення. Припускається рівень впливу навіть до заборони впливу деяких із них.

У більшості ситуацій функції активації знижують значення або мають меншу за діапазон вхідних даних, з великою ймовірністю обмежену область значень. Це пов'язано з тим, що в інакшому випадку існує ризик нескінченного зростання параметрів. Також вони мають бути диференційовними. Це потрібно для уможливлення алгоритму зворотнього поширення помилки. Відповідно, стають можливими градієнтні методи навчання мережі. Вибір функції активації має значний вплив на спосіб, у який потрібно робити вибір початкових значень для нейронної мережі перед початком її навчання. Прикладом може слугувати ситуація, коли функція активації не наближує тотожне перетворення поблизу нульової точки, то у результаті задання значень мережа може гірше навчатися в майбутньому.

Випрямлена лінійна одиниця (англ. rectified linear unit, скорочення ReLU) – це функція активації, що дедалі частіше викривується в сучасних нейронних мережах. Розповсюдженість саме такої функції пояснюється тим, що вона

оброблює дані та навчає мережу у декілька разів швидше, ніж це роблять інші функції активації. У той час також створюється нелінійність. Цей фактор, окрім пришвидшення навчання мережі, ще зменшує вірогідність її перенавчання. Це зумовлено тим, що мережа навчається більш складним зв'язкам за однаковий період часу, у порівнянні з іншими функціями активації, що в свою чергу дозволяє навчати складніші моделі, використовуючи ті самі дані. Формула знаходження значення виглядає наступним чином:

$$y = f(x) = \max(0, x) \quad (2.8)$$

де  $y$  – елемент поточного рівня,  $x$  – елемент з вхідних даних.

Як можна зрозуміти із формули функції, вона усуває негативні значення з мережі, надаючи таким чином їм нульового значення. Натомість вона виконує тотожне перетворення для інших значень. Швидкість роботи цієї функції з апаратної точки зору підвищується також завдяки тому, що вона є звичайним порівнянням значення з нулем. Похідна ReLU дорівнює 0 для від'ємних значень і 1 для всіх інших, тому процес роботи алгоритму зворотнього поширення помилки також стає спрощеним. Тим не менше, ReLU не наближує тотожного перетворення поблизу нульової точки. Саме тому вона потребує наявності хоча б одного додатного початкового значення для ефективного навчання. Проте слід зауважити, що активація не всіх початкових значень позитивно впливає на роботу мережі через ефект розрідження. Впровадження ReLU було пов'язано зі знайденням аналогії з її роботою в обробці візуальної інформації корою головного мозку.

Ще двома функціями, що традиційно застосовуються для функцій активації в нейронних мережах, є гіперболічний тангенс ( $f(x) = \tanh(x)$ ) та функція сігмоїди, що також використовується в якості функції активації на останньому рівні мереж в задачах бінарної класифікації. Це зумовлено тим, що її область значень дорівнює  $[0; 1]$ . Це дає можливість апроксимувати імовірності належності вхідних даних до двох класів:

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (2.9)$$

В задачах багатокласової класифікації на останньому рівні мережі у більшості випадків застосовується функція Softmax завдяки тому, що вона враховує всі значення з попереднього рівня мережі та може обрахувати відповідну імовірність для кожного з класів. Саме ця функція використовується для формування результатів в цій роботі:

$$y_i = f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}, \forall i, j \in (1, N) \quad (2.10)$$

де  $y_i$  – елемент поточного шару з індексом  $i$ ,

$x_i$  – елемент попереднього шару з індексом  $i$ ,

$N$  – кількість елементів на попередньому шарі.

Фактично, функція Softmax нормалізує експоненційні значення даних з попереднього рівня, тим самим переводячи вхідний масив довільних значень в область  $[0;1]$ . Цей процес використовується для обчислення імовірностей. Функція може бути модифікована за допомогою заміни бази степені з  $e$  на інше число. Це дає можливість збільшити вагу великих значень на попередньому рівні (при збільшенні бази), або навпаки зрівнюючи вплив різних значень (при зменшенні бази). Також функція стійка до збільшення всіх показників з попереднього рівня на сталі значення за рахунок збільшення стійкості результуючих імовірностей належності до класів.

## Висновки до розділу 2

В даному розділі було розглянуто поняття згорткових нейронних мереж, а також один з підвидів нейронних мереж. Він демонструє свою ефективність в задачах класифікації та розпізнавання зображень, але також може бути використаним також для інших задач. Головними елементами ЗНМ є згорткові рівні з операцією згортки. Вони розпізнають ознаки на зображення, агрегувальні рівні з операціями, які узагальнюють зображення та забезпечують покращену стійкість моделі, повнозв'язні рівні для обробки результатів, отриманих на згорткових та агрегувальних, рівні втрат, що забезпечують оновлення параметрів моделі при навчанні та покращення її роботи, та функції активації, які забезпечують нелінійність системи та роблять можливими багат шарову побудову мереж, а також розпізнавання складних характеристик.

Згорткові нейронні мережі навчаються за допомогою градієнтних алгоритмів, які змінюють параметри моделей, забезпечуючи зменшення помилки. Для задач класифікації, що детально розбираються у цій роботі, результатом роботи згорткової нейронної мережі стає імовірності належності вхідного зображення до кожного з вибраних класів.



## **3 МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ**

Коли перед розробниками ставиться задача із застосування статистичних методів або аналізу даних у своїй роботі, вибір нерідко робиться на користь мови Python. Вона також широко застосовується вченими, чиї завдання мають бути інтегрованими із веб додатками або виробничими середовищами.

### **3.1 Вибір програмних засобів**

Для вирішення описаної у попередніх розділах задачі було обрано мову програмування Python. Вона є оптимальною у даній ситуації, оскільки широко застосовується для організації машинного навчання. Це обумовлено тим, що кодувати даною мовою відносно просто легко та швидко. У той час, як для реалізації ідентичної функціональності на мові C++ або Java необхідно було б написати набагато значно більшу кількість строк коду. Крім того, мова Python відрізняється простим синтаксисом та наявністю динамічної типізації. При застосуванні нейронних мереж необхідно проводити велику кількість експериментів. Серед них: визначення архітектури нейронної мережі, яка буде підходити для поставленої задачі, оцінка параметрів навчання тощо. Для описаних цілей мова Python є актуальною, оскільки дає можливість легко модифікувати програму для проведення різноманітних експериментів, При цьому програма при таких модифікаціях залишає свої властивості, а саме: легкість читання та високу якість підтримки.

Одну перевагу Python потрібно виділити окремо. Вона є головною причиною застосування мови для машинного навчання. Перевага полягає в наявності великої кількості фреймворків і бібліотек. Вони суттєво полегшують процес написання коду і зменшують кількість часу, що витрачається на розробку. До основних фреймворків і бібліотек, що застосовуються у машинному навчанні, а також будуть використані у даній роботі належать наступні:

- NumPy;

- SciPy;
- Scikit-learn;
- Matplotlib;
- Pandas;
- Tensor Flow;
- Keras.

NumPy – це головна бібліотека Python. Вона полегшує роботу із векторами та матрицями. Бібліотека містить готові методи для операцій різних типів. Завдяки NumPy розробники проводять створення, зміни форми, множення і розрахунок детермінанту матриць, вирішують лінійні рівняння, проводять сингулярне розкладання, а також виконують безліч інших завдань.

SciPy – це бібліотека, призначена спеціально для наукових розрахунків. Вона базується на NumPy та значно розширює її можливості. До неї входять методи лінійної алгебри разом із методами, призначеними для роботи з ймовірнісними розподілами, інтегральним обчисленнями, а також із перетвореннями Фур'є.

Scikit-learn – бібліотека, що базується на NumPy і SciPy. Вона забезпечує програміста алгоритмами для машинного навчання, а також інтелектуального аналізу наступних категорій даних: кластеризації, регресії, а також класифікації.

Бібліотека Matplotlib відноситься до низькорівневих, вона призначена для створення двовимірних діаграм і графіків. Одна із важливих задач машинного навчання – це візуалізація даних. Використовуючи бібліотеку Matplotlib, можна побудувати графік будь-якого типу.

Pandas – бібліотека, що забезпечує розробників структурами даних і інструментами для аналізу. Може застосовуватися для обробки неповних, неупорядкованих і немаркованих даних (які найбільш часто можуть зустрітися в житті). Pandas дає можливість замінити досить складні операції з даними однією чи двома командами. До її складу входить велика кількість готових методів групування, фільтрації, об'єднання даних, а також можливість розпізнавання різноманітних джерел. У Pandas є можливість об'єднувати таблиці аналогічно до

SQL JOIN. При цьому дані беруться безпосередньо з файлів. Завдяки цій системі зникає необхідність організувати бази даних. Додаткова перевага бібліотеки полягає у швидкості роботи.

TensorFlow – це бібліотека, що застосовується у процесі роботи із штучними нейронними мережами з численними наборами даних. Завдяки їй відбувається налаштування, тренування і застосування зазначених мереж. Дана бібліотека машинного навчання розроблена компанією Google. Корпорація на сьогодні є одною з передових компаній, що застосовують на практиці машинне навчання. TensorFlow відноситься до бібліотек із відкритими вихідними кодами. Два основних компоненти, зазначені у назві – це проведення операцій із багатовимірними матрицями (тензорами), що широко використовуються у процесі навчання нейронних мереж, а також розрахунки на графах потоків даних (flow) між якими відбувається передача тензорів. Один із недоліків TensorFlow полягає у необхідності описувати замість нейронної мережі граф потоків даних, що відповідає релевантній нейронній мережі. Це означає, що існує необхідність самостійно реалізувати нейронну мережу поверх ефективних обчислень з тензорами.

Враховуючи викладену вище інформацію, задля розробки програми буде застосована ще одна допоміжна бібліотека глибокого навчання – Keras. Вона застосовує TensorFlow для виконання ефективних обчислень. Одна з особливостей бібліотеки Keras полягає в тому, що бібліотека дає можливість описувати нейронну мережу на Python. За допомогою Keras розробник отримає можливість вказати шари, з яких складатиметься нейронна мережа, функції активації, що будуть застосовуватись, метод оптимізації для зменшення помилки та інші параметри, які мають велике значення для навчання нейронної мережі. Далі Keras самостійно побудує необхідну нейронну мережу і для виконання обчислень буде викликати високоефективні методи з TensorFlow.

Традиційні програми, написані мовою Python працюють досить повільно, проте бібліотеки глибокого навчання використовують різноманітні методи для

підвищення швидкості роботи програм. Так, бібліотеки використовують оптимізовані математичні бібліотеки, створені на C. Деякі з бібліотек, засновані на кодї мовою Python автоматично генерують код мовою C. У подальшому виконується вже цей код, який працює значно швидше. Для ще більш суттєвого збільшення рівня продуктивності припускається використання графічних прискорювачів (GPU) із використанням NVIDIA cuDNN.

Для створення коду було застосовано інтерактивну оболонку, призначену для мови Python – Jupyter Notebook. Вона забезпечує розширену інтроспекцію, наявність додаткового командного синтаксису, наявність доступу до оболонки системи, взаємодію із pdb відладчиком і Python профайлером, а також забезпечує збереження історії вводу даних, у всіх сеансах, підсвічує код здійснює автоматичне його доповнення. Jupyter Notebook відрізняється можливістю розробляти, документувати та запускати програми, написані мовою Python. До його складу входять 2 компоненти: веб-додаток, що відкривається у браузері, і ноутбуки. Notebooks називаються файли, у яких передбачена можливість роботи із вихідним кодом програми, запускати цей код, здійснювати введення та виведення даних тощо. Веб додаток забезпечує розробника можливістю для редагування Python коду у браузері. При цьому відбувається підсвічування синтаксису, автовідступ та автодоповненням. Також є можливість відкривати код в браузері, виводити результати обчислень з медіа поданням (схеми, графіки). Ноутбуки – це файли, які містять вихідний код, вхідні і вихідні дані, що були отримані в процесі сесії. Фактично, він являє собою запис роботи. При цьому є можливість заново виконати код, записаний на ньому. Є можливість експортувати ноутбуки у формати PDF і HTML. Jupyter Notebook входить до складу Anaconda. Це дистрибутив для мови Python.

Крім того, у процесі експериментального моделювання було застосовано Google Colaboratory. Це безкоштовна платформа для машинного навчання і нейронних мереж . На ній вже встановлено усі необхідні бібліотеки для навчання нейронних мереж: Keras, TensorFlow, PyTorch та інші. На даній платформі можна

використовувати ноутбуки, що мають багато спільного із ноутбуками Jupyter. Головна перевага Google Colaboratory в тому, що навчання нейронних мереж вимагає великих об'ємів ресурсів для обчислення, а Google Colaboratory безкоштовно забезпечує користувачів досить потужним GPU. Таким чином, використовуючи платформу Google Colaboratory, вдалося суттєво підвищити швидкість навчання нейронних мереж.

З метою здійснити автоматичний підбір гіперпараметрів нейронної мережі застосовувався інструмент Keras Tuner. Це оптимізатор гіперпараметрів, від компанії Google, спеціально для Keras у складі Tensorflow 2.0. Усі гіперпараметри мережі взаємопов'язані. Саме тому підібрати вручну порібну комбінацію гіперпараметрів, при якій використовується максимально можлива якість роботи нейронної мережі, непросто. За допомогою інструменту Keras Tuner вдалося підібрати потрібну об'єм прихованих шарів у мережі, число нейронів у кожному шарі, функцію активації, а також оптимізатор.

### **3.2 Підготовка набору даних для навчання**

Однією з основних і найбільш болючих проблем машинного навчання та науки про дані є самі дані, точніше їх відсутність. Незважаючи на те, якими великими кроками рухається вперед технічний прогрес, в усьому світі, зокрема в Україні, досі є петабайти даних, що не оцифровані і зберігаються в архівах, без можливого онлайн доступу до них.

Збір даних є найбільш затратним по часу у машинному навчанні та є актуальною темою досліджень у багатьох спільнотах. Існує в значній мірі дві причини збору даних, які останнім часом стали критичним питанням. По-перше, оскільки машинне навчання стає все більш широко використовуваним, ми бачимо нові можливості використання, які не обов'язково мають достатньо позначених даних. По-друге, на відміну від традиційного машинного навчання, де інженерна особливість є вузьким місцем, глибокі методи навчання автоматично генерують особливості, але замість цього вимагають великих обсягів позначених даних.

Збір даних дозволяє зафіксувати минулі події, щоб ми могли використовувати аналіз даних для пошуку повторюваних моделей. З цих моделей створюються моделі прогнозування, використовуючи алгоритми машинного навчання, які шукають тенденції та передбачають майбутні зміни.

Ефективність моделей прогнозування залежить від даних, на основі яких вони побудовані, тому вирішальне значення має вибір оптимальних практик збору даних для моделей. Усі дані повинні бути валідними, відповідати поставленій задачі.

З метою здійснення навчання нейронної мережі було застосовано набір даних Chest CT-ray Images. Він був сформований із реальних даних пацієнтів працівниками Гуанчжоуського жіночого та дитячого медичного центру у Китаї. Цей масив даних знаходиться у відкритому доступі та містить 5863 файли (загальний розмір: 2Гб). До набору даних входять рентгенівські та знімки КТ у форматі JPEG, що розділені на 2 класи (COVID / норма). Застосований масив даних складається з трьох каталогів (train, test, val) і включає в себе підкаталоги для кожної з категорій зображень (COVID / Normal).

Всі КТ-знімки органів грудної клітки були отримані у ході стандартного надання клінічної допомоги пацієнтам. Автори цього набору провели контроль якості зображень. Були відбраковані низькоякісні знімки або такі, що неможливо зчитати. Перед тим, як знімки були відібрані для навчання системи ШІ, діагнози відповідних пацієнтів були перевірені двома висококваліфікованими лікарями. З метою виключення будь-яких помилок в оцінюванні, ця інформація була потім перевірена декількома спеціалістами.

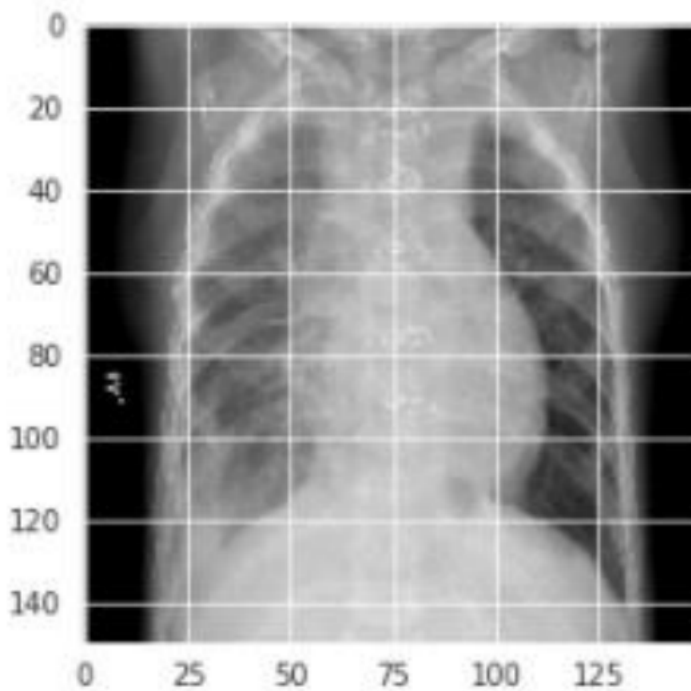


Рис. 3.1. Приклад зображення з набору даних Chest CT-ray Images

Для здійснення навчання нейронної мережі масив даних був розділений на дві частини: навчальну (training set) та тестову (test set). Співвідношення вибірок – 80:20 відповідно. Наступним кроком було виділення з навчального набору даних 10% на перевіірочну вибірку (validation set) – масив даних, що буде застосовуватись в процесі навчання для підбору гіперпараметрів мережі.

У результаті до навчальної вибірки увійшло 5000 зображення, до тестової 1350, а до перевіірочної 560 знімків. Далі набір даних було перемішано з метою забезпечення рівномірного представлення кожного із класів в усіх трьох вибірках. Можна побачити, що у масиві даних простежується нерівномірний розподіл знімків за класами. Видно, що у навчальну вибірку увійшло 4250 зображення із класом COVID та 750 зображення класу Норма, у тестову вибірку – 907 зображення COVID і 443 Норми, а до перевіірочної вибірки потрапили 420 і 140 зображень із кожним класом відповідно. Можна зрозуміти, що у кожній вибірці співвідношення класів дорівнює приблизно 3:1. З урахуванням такої нерівномірності даних, а також їх порівняно невеликої кількості, якої не вистачить для якісного навчання мережі,

було використано техніку доповнення даних (англ. data augmentation). Результат розподілу знімків за класами у трьох вибірках відображений у таблиці 3.1.

Таблиця 3.1

### Розподіл зображень вибірок

| Результат | Навчальна (training set) | Тестова вибірка (test set) | Перевірочна вибірка (validation set) |
|-----------|--------------------------|----------------------------|--------------------------------------|
| Норма     | 750                      | 443                        | 140                                  |
| COVID     | 4250                     | 907                        | 420                                  |
| Всього    | 5000                     | 1350                       | 560                                  |

Доповнення даних може суттєво збільшити точність навчання нейронної мережі. Це дає високі результати, якщо кількість даних недостатня для навчання, вони не є збалансованими, а також відсутнє рівномірне розподілення різних класів. Така тенденція і спостерігається у даному наборі даних. З метою доповнення даних було застосовано функцію генератора зображень (ImageGenerator), яку містить бібліотека глибокого навчання Keras. У процесі налаштуванні генератора було задано наступні можливості:

- поворот знімку кожному із можливих напрямках зі значенням до 30°;
- зсув по ширині на 10% від загального значення на знімку або менше;
- зсув по висоті на 10% від загальної висоти знімку чи менше;
- підвищення розміру знімку на 20% від його початкового розміру або менше;
- відображення знімку по горизонталі;
- відображення знімку по вертикалі.

Це означає, що в процесі використання створеного генератора, Keras буде випадково обирати декілька варіантів модифікації зображення і застосовувати їх до знімків із масиву даних.

Для тестового та перевірного масивів даних доповнення даних не застосовується. Це зумовлено тим, що більш коректно оцінювати навчену мережу



на основі незмінених даних – у більшості ситуації можна побачити підвищення точності на тестових даних. Існує ймовірність, що така тенденція спостерігається за рахунок незначного зниження рівня точності на навчальних зображеннях. Перед подачею на вхід нейронної мережі необхідна нормалізація даних. Нейронна мережа має можливість приймати на вхід лише вектори з однаковою розмірністю. Зважаючи на це, за допомогою бібліотеки OpenCV всі зображення були стандартизовані до єдиного розміру – 150x150 пікселів. Наступним кроком була здійснена нормалізація даних про інтенсивність пікселів у зображенні – значення кожного з пікселів було розділено на 255. Це було зроблено для того, щоб дані на вході у нейронну мережу перебували у діапазоні від 0 до 1. Це полегшує роботу алгоритмів оптимізації, що застосовуються для навчання нейронних мереж.

### 3.3 Модифікація алгоритму

Доцільно розробити алгоритм генерації хеш-коду, виходячи з переваг розбиття зображення на частини. Розроблений алгоритм має схожі риси з алгоритмом Average Hash, який обчислює середні значення пікселів для обчислення хеш-коду. В модифікованому варіанті зображення ділиться на  $N$  частин. Далі, для модифікованого методу середні значення рахуються для кожного пікселя (всього зображення), а для пікселів кожного блоку відповідно. Як результат отримаємо  $N$  значень середніх для кожної частини. Для генерування результуючого хеш-коду є можливим порівняння середніх значень кожної частини з їх значенням медіани. Це дозволяє отримати бінарний результат хеш-коду. Опишемо алгоритм.

Нехай  $N$  позначає довжину хеш-коду (в бітах).  $k$  – кількість значень пікселів у кожній частині у результаті розбиття зображення на частини:

- зменшити зображення та інтерполювати. Наприклад, слідуючи BTC алгоритму, можна зменшити розмір зображення до  $N*4 \times N*4$ . Цей розмір шматків та виконання наступних кроків алгоритму отримають  $N$  блоків розміром  $4 \times 4$ ;
- перевести зображення у градації сірого. Як результат створимо матрицю значень відтінків градацій сірого;

- розбити зображення  $I$  на  $N$  блоків та розгорнути їх у вектори  $\{I_1, I_2, \dots, I_N\}$ , що не перетинаються. На практиці, довжина хеш-коду повинна становити хоча б 64 біти, тому на практиці кількість блоків становить 64 та більше;
- Обчислити середнє значення кожного вектора. Тобто, отримаємо послідовність середніх значень  $\{M_1, M_2, \dots, M_N\}$  відповідних векторів  $\{I_1', I_2', \dots, I_N'\}$ ;
- Обчислити медіанне значення  $M_d$  всіх середніх значень  $\{M_1, M_2, \dots, M_N\}$ . Медіана — це величина ознаки, що розташована посередині ранжованого ряду вибірки, тобто це — величина, що розташована в середині ряду величин у зростаючому або спадному порядку. Медіана ділить ряд значень ознаки на дві рівні частини, по обидві частини від неї розміщується однакова кількість одиниць сукупності. Тобто, вектор значень  $\{M_1, M_2, \dots, M_N\}$  сортується за зростанням, а потім знаходиться індекс медіанного значення;
- звести середні значення  $\{M_1, M_2, \dots, M_N\}$  у бінарний вигляд та отримати хеш-код  $h$ .

В основі запропонованого алгоритму є алгоритм Average Hash та алгоритм компресії зображень під назвою Block Truncation Coding. Основною ідеєю є розбиття зображень на блоки та обчислення хеш-кодів на основі кожного блоку окремо. Такий підхід є ефективним у плані часу виконання забезпечує часткову інваріантність до обрета зображень. Також, розглянуті методи інтерполяції зображення та методи зведення зображень у градації сірого. Інтерполяція є важливим кроком при зменшенні розміру зображення, що використовується в алгоритмі. Визначено, що метод бікубічної інтерполяції є оптимальним по співвідношенню часу обробки зображення та якістю результату. Тому використання цього методу інтерполяції є доцільним для роботи алгоритму та модифікації зображень при його тестуванні.

Існує декілька методів зведення зображень до градацій сірого. Оптимальним зі сторони людського сприйняття є метод під назвою «Luminosity». Розроблений алгоритм зводить зображення до градацій сірого для подальших обчислень. Фактор

людського сприйняття не є важливим для даного алгоритму. Тому для запропонованого алгоритму доцільно використовувати метод під назвою «Average». Для порівняння томографічних зображень даний крок не є необхідним, адже такі зображення подаються у чорно-білому вигляді. Але, для подальших досліджень, тестування алгоритму та його аналізу є доцільним використання різних зображень, у тому числі і кольорових.

### 3.4 Аналіз результатів

Фінальна підсистема була реалізована за допомогою мови програмування Python і застосуванню бібліотек машинного навчання TensorFlow та Keras.

Така підсистема дає можливість завантажити знімок, що буде класифікований нейронною мережею і з певною долею ймовірністю віднесений до будь-якого із двох класів – COVID або Норма.

Описана у роботі мережа є глибокою. Саме тому на проведення повного її навчання (впродовж 30 епох) на машині без GPU знадобилося більше, ніж 15 годин на повний процес навчання мережі. Даний результат був можливий за рахунок невеликого обсягу вхідних даних, а також застосування ЗНМ. Вони дають знизити кількість вагових коефіцієнтів, які потрібно обчислити на кожній епосі. Відповідно, такий підхід скорочує час навчання.

Наприкінці навчання мережі точність на даних, на яких проводилося навчання, дорівнювала 95.5%, а на перевіірочній вибірці (validation data) – 95.9%. Значення функції втрат (loss) відповідно дорівнювали 0.14 на навчальній вибірці та 0.16 на перевіірочній вибірці (validation loss) (рисунок 3.2).

Кафедра інтелектуальних інформаційних систем  
 Діагностування COVID-19 на основі методів машинного навчання

```

Epoch 1/30
132/132 [=====] - 26s 200ms/step - loss: 1.5610 - accuracy: 0.7752 - val_loss: 4.4258 - val_accuracy: 0.7569
Epoch 2/30
132/132 [=====] - 26s 193ms/step - loss: 0.3990 - accuracy: 0.8411 - val_loss: 1.5706 - val_accuracy: 0.7569
Epoch 3/30
132/132 [=====] - 26s 193ms/step - loss: 0.3510 - accuracy: 0.8742 - val_loss: 6.8769 - val_accuracy: 0.7655
Epoch 4/30
132/132 [=====] - 25s 193ms/step - loss: 0.3385 - accuracy: 0.8868 - val_loss: 0.5616 - val_accuracy: 0.7569
Epoch 5/30
132/132 [=====] - 26s 195ms/step - loss: 0.2753 - accuracy: 0.9005 - val_loss: 1.0836 - val_accuracy: 0.8316
Epoch 6/30
132/132 [=====] - 26s 193ms/step - loss: 0.2699 - accuracy: 0.9086 - val_loss: 0.4880 - val_accuracy: 0.7591
Epoch 7/30
132/132 [=====] - 25s 193ms/step - loss: 0.3041 - accuracy: 0.9088 - val_loss: 0.5063 - val_accuracy: 0.8742
Epoch 8/30
132/132 [=====] - 26s 194ms/step - loss: 0.2337 - accuracy: 0.9138 - val_loss: 1.7783 - val_accuracy: 0.7868
Epoch 9/30
132/132 [=====] - ETA: 0s - loss: 0.2684 - accuracy: 0.9105
Epoch 0009: ReduceLROnPlateau reducing learning rate to 0.0003000000142492354.
132/132 [=====] - 26s 194ms/step - loss: 0.2684 - accuracy: 0.9105 - val_loss: 0.3961 - val_accuracy: 0.8401
Epoch 10/30
132/132 [=====] - 26s 196ms/step - loss: 0.1808 - accuracy: 0.9389 - val_loss: 1.5576 - val_accuracy: 0.7569
Epoch 11/30
132/132 [=====] - ETA: 0s - loss: 0.1801 - accuracy: 0.9441
Epoch 0011: ReduceLROnPlateau reducing learning rate to 9.000000427477062e-05.
132/132 [=====] - 26s 196ms/step - loss: 0.1801 - accuracy: 0.9441 - val_loss: 76.6414 - val_accuracy: 0.7676
Epoch 12/30
132/132 [=====] - 26s 196ms/step - loss: 0.1871 - accuracy: 0.9479 - val_loss: 1.5775 - val_accuracy: 0.9275
Epoch 13/30
132/132 [=====] - 26s 195ms/step - loss: 0.1765 - accuracy: 0.9505 - val_loss: 0.1586 - val_accuracy: 0.9446
Epoch 14/30
132/132 [=====] - 26s 196ms/step - loss: 0.1473 - accuracy: 0.9517 - val_loss: 0.1013 - val_accuracy: 0.9595
Epoch 15/30
132/132 [=====] - 26s 195ms/step - loss: 0.1333 - accuracy: 0.9543 - val_loss: 1.3486 - val_accuracy: 0.9147
Epoch 16/30
132/132 [=====] - ETA: 0s - loss: 0.1313 - accuracy: 0.9538
Epoch 0016: ReduceLROnPlateau reducing learning rate to 2.700000040931627e-05.
132/132 [=====] - 26s 197ms/step - loss: 0.1313 - accuracy: 0.9538 - val_loss: 1.2234 - val_accuracy: 0.9275
Epoch 17/30
132/132 [=====] - 26s 195ms/step - loss: 0.1373 - accuracy: 0.9545 - val_loss: 0.1461 - val_accuracy: 0.9467
Epoch 18/30
132/132 [=====] - ETA: 0s - loss: 0.1353 - accuracy: 0.9505
Epoch 0018: ReduceLROnPlateau reducing learning rate to 8.100000013655517e-06.
132/132 [=====] - 26s 197ms/step - loss: 0.1353 - accuracy: 0.9505 - val_loss: 0.1365 - val_accuracy: 0.9446
Epoch 19/30
132/132 [=====] - 26s 197ms/step - loss: 0.1381 - accuracy: 0.9557 - val_loss: 0.1361 - val_accuracy: 0.9467
Epoch 20/30
132/132 [=====] - ETA: 0s - loss: 0.1249 - accuracy: 0.9512
Epoch 0020: ReduceLROnPlateau reducing learning rate to 2.429999949526973e-06.
132/132 [=====] - 26s 195ms/step - loss: 0.1249 - accuracy: 0.9512 - val_loss: 0.1707 - val_accuracy: 0.9531
Epoch 21/30
132/132 [=====] - 26s 196ms/step - loss: 0.1409 - accuracy: 0.9522 - val_loss: 0.1566 - val_accuracy: 0.9488
Epoch 22/30
132/132 [=====] - ETA: 0s - loss: 0.1396 - accuracy: 0.9564
Epoch 0022: ReduceLROnPlateau reducing learning rate to 1e-06.
132/132 [=====] - 26s 196ms/step - loss: 0.1396 - accuracy: 0.9564 - val_loss: 0.1872 - val_accuracy: 0.9552
Epoch 23/30
132/132 [=====] - 26s 196ms/step - loss: 0.1356 - accuracy: 0.9548 - val_loss: 0.1164 - val_accuracy: 0.9616
Epoch 24/30
132/132 [=====] - 26s 199ms/step - loss: 0.1446 - accuracy: 0.9533 - val_loss: 0.1280 - val_accuracy: 0.9510
Epoch 25/30
132/132 [=====] - 27s 203ms/step - loss: 0.1491 - accuracy: 0.9557 - val_loss: 0.1648 - val_accuracy: 0.9467
Epoch 26/30
132/132 [=====] - 26s 199ms/step - loss: 0.1202 - accuracy: 0.9567 - val_loss: 0.1160 - val_accuracy: 0.9574
Epoch 27/30
132/132 [=====] - 26s 198ms/step - loss: 0.1370 - accuracy: 0.9529 - val_loss: 0.1402 - val_accuracy: 0.9595
Epoch 28/30
132/132 [=====] - 26s 198ms/step - loss: 0.1467 - accuracy: 0.9552 - val_loss: 0.1399 - val_accuracy: 0.9510
Epoch 29/30
132/132 [=====] - 26s 196ms/step - loss: 0.1444 - accuracy: 0.9545 - val_loss: 0.1783 - val_accuracy: 0.9382
Epoch 30/30
132/132 [=====] - 26s 197ms/step - loss: 0.1353 - accuracy: 0.9557 - val_loss: 0.1558 - val_accuracy: 0.9595

```

Рис. 3.2. Вивід бібліотеки Keras в процесі навчання мережі

Було створено графіки зміни помилки (рис. 3.3), а також точності мережі у ході навчання (рис 3.4). На цих графіках можна побачити, що у процесі навчання мережі на кожній з епох рівень помилки на наборі даних для навчання і наборі даних для перевірки стає меншим, а показник якості навчання збільшується. Це означає, що перенавчання мережі не виникло. На перших епохах навчання помилка

падає як на даних для навчання, так і на перевірочних. А ближче до завершення навчання рівень помилки знижується лише на даних для навчання, а на наборі даних для перевірки значення помилки практично залишається сталим. Крім того, на наборі даних для перевірки майже не росте показник якості навчання. На основі цієї інформації можна зробити висновок, що якщо продовжити навчати мережу, її узагальнююча здатність буде знижуватися. Мережа стане гірше розпізнавати нові зображення, з якими не контактувала в процесі навчання. Тому процес навчання мережі було припинено, щоб не відбулося перенавчання.

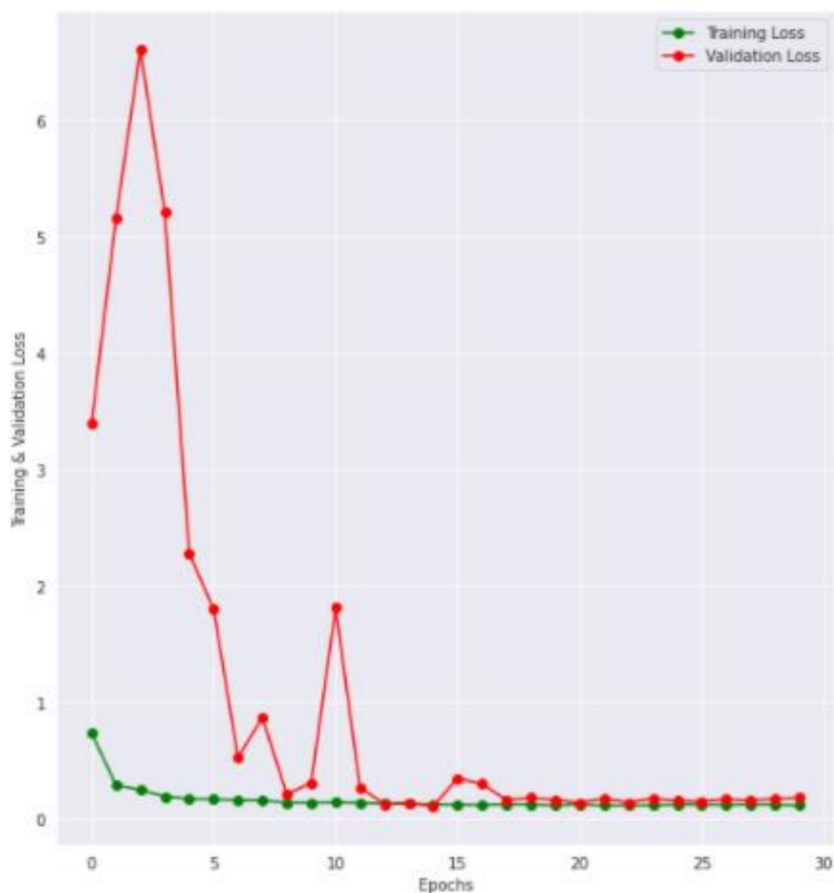


Рис. 3.3. Графік помилки моделі

Кафедра інтелектуальних інформаційних систем  
Діагностування COVID-19 на основі методів машинного навчання

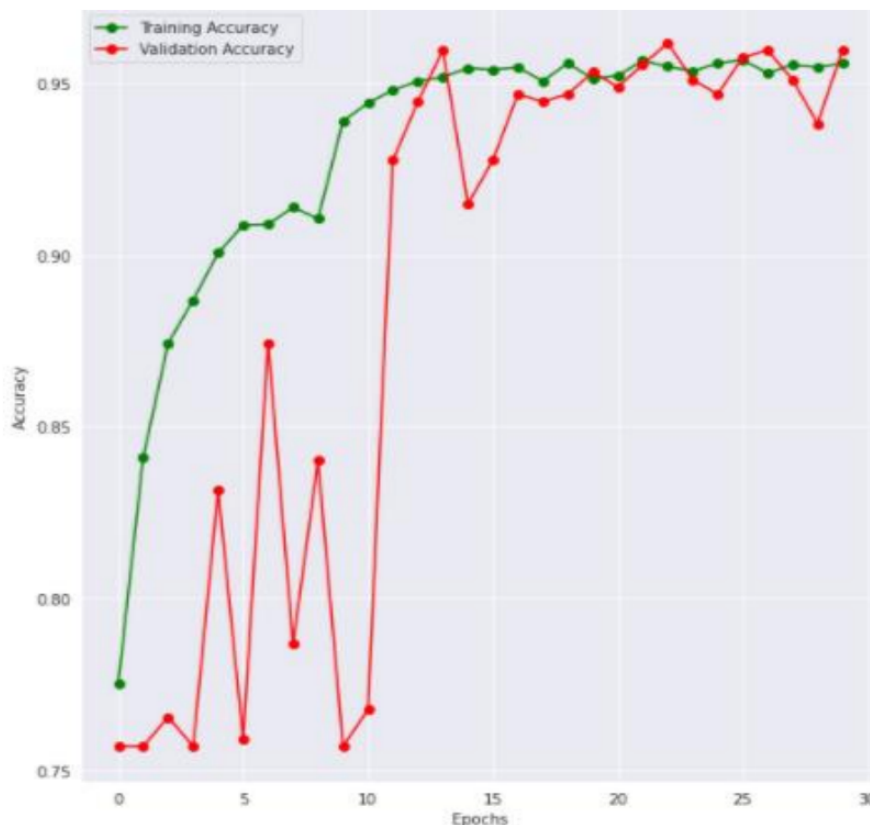


Рис. 3.4. Графік точності моделі

Після завершення процесу навчання була здійснена перевірка якості роботи мережі на тестових даних, які мережа не ідентифікувала у процесі навчання (рис. 3.5). Дані цього типу були заздалегідь відібрані з навчального масиву і становлять 1172 знімків. Для цієї конфігурації мережі і обраних параметрів навчання рівень точності склав 93.4%. Це менший показник, аніж на тестовій і навчальній вибірці. Проте цей рівень точності є вищим за точність аналогічних систем на 0.7% – найбільш точний із досліджених аналогів демонструє точність у 92.7%. Крім того, вона є вищою, за рівень точності в процесі розпізнавання пневмонії на рентгенівських знімках звичайною людиною та навіть кваліфікованими лікарями-радіологами. Цей рівень за даними досліджень складає всього 82.8%. Рівень похибки на тестових даних становила 0.2 що є не набагато більше, а ніж для результатів на навчальних та перевіірочних даних. Таким чином, продемонстровані системою показники точності та помилки можна вважати цілком задовільними.

```
37/37 [=====] - 1s 30ms/step - loss: 0.1987 - accuracy: 0.9343
Loss of the model is - 0.19866324961185455
37/37 [=====] - 1s 23ms/step - loss: 0.1987 - accuracy: 0.9343
Accuracy of the model is - 93.43003630638123 %
```

Рис. 3.5. Значення точності та помилки на тестових даних

Було здійснено порівняння якості роботи мережі на тестових даних за умови застосування техніки доповнення даних (англ. data augmentation) та без такої. Доповнення використовувались лише з метою навчання мережі, оскільки вже навчену мережу потрібно оцінювати, спираючись на немодифіковані дані тестування. Без застосування даної техніки мережі знадобилось менше часу на здійснення навчання – всього 11 епох і 15 секунд на кожну з них. Проте рівень точності на тестових склав всього 84.7%. Тим часом помилка складала 4.4. Таким чином було виявлено, що використання data augmentation для цього масиву даних дало перевагу у точності у 9.21% і менший у 22 рази рівень помилки. Проте процес навчання зайняв на понад 25 хвилин більше часу.

Крім того, було оцінено інші характеристики якості навчання мережі. Зокрема аналізувалися такі параметри:

- точність (precision);
- відгук (recall);
- f-міра (f-measure).

Значення цих параметрів за класами можна побачити на рисунку 3.6. Можна побачити, що модель більш успішно розпізнає зображення коронавірусу, тому що знімків цього класу було у наявності більше у навчальній вибірці. Але в цілому значення цих метрик може бути оцінено як високе.

|           | precision | recall | f1-score | support |
|-----------|-----------|--------|----------|---------|
| (Class 0) | 0.94      | 0.97   | 0.95     | 804     |
| (Class 1) | 0.92      | 0.86   | 0.89     | 368     |
| accuracy  |           |        | 0.93     | 1172    |
| macro avg | 0.93      | 0.91   | 0.92     | 1172    |
| ghted avg | 0.93      | 0.93   | 0.93     | 1172    |

Рис. 3.6. Значення мір якості навчання за класами

Число коректно та некоректно класифікованих зображень у ході тестування мережі за класами видно на рисунку 3.7. На рис. 3.8 видно декілька зображень, були вірно віднесені мережею до класу Норма, а на рис. 3.9 – зображення класу Норма, які мережею були невірно класифіковані як COVID.

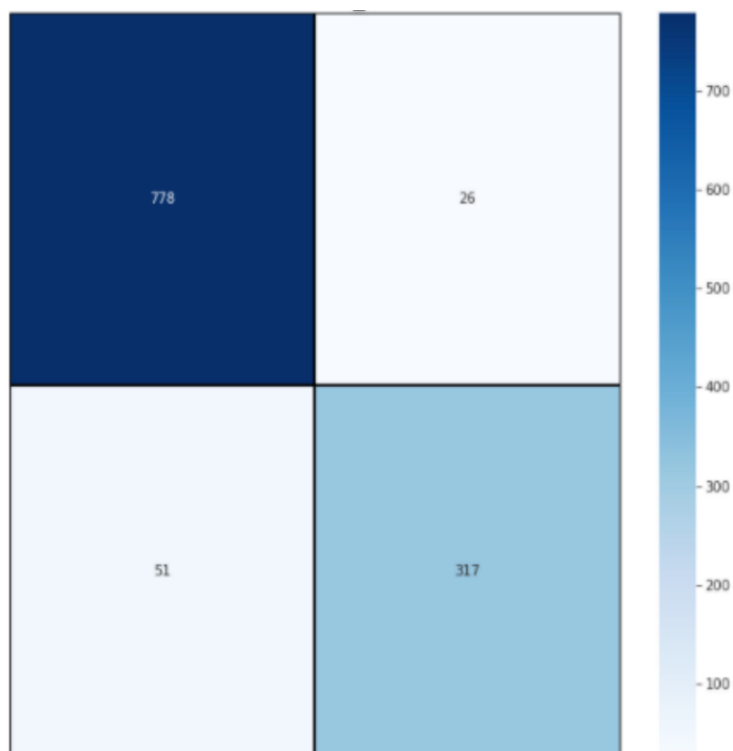


Рис. 3.7. Розподіл відповідей мережі за класами

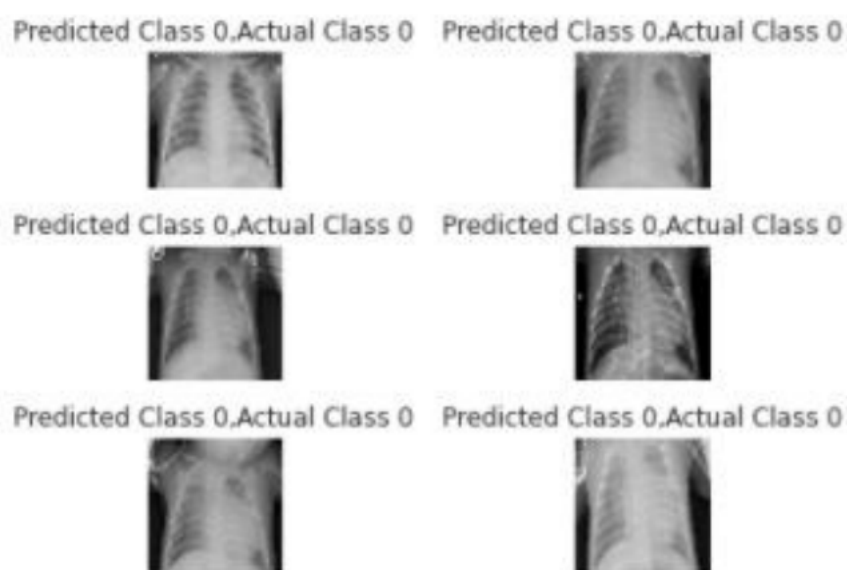


Рис. 3.8. Приклади правильно класифікованих зображень



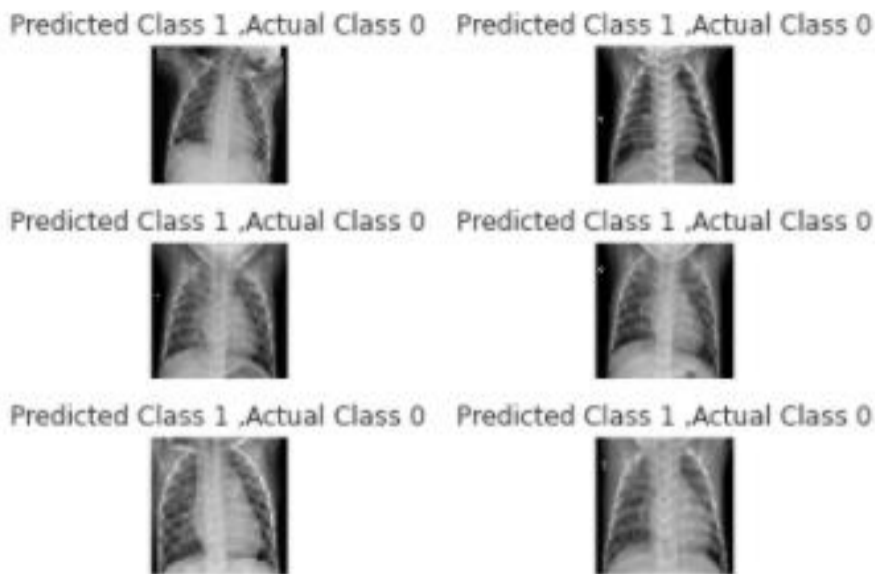


Рис. 3.9. Приклади неправильно класифікованих зображень

Крім того, було здійснено порівняння рівня точності роботи на тестових даних побудованої моделі із попередньо навченими мережами. Це було зроблено з метою визначення доцільності побудови власної моделі. З цією метою було відібрано наступні найбільш популярні мережі з модуля Keras Applications, що застосовується для розпізнавання об'єктів на зображеннях: VGG16, VGG19 та U-NET. Ці мережі було додатково навчено на навчальних даних з масиву Chest CT-ray Images. З цією метою було введено класифікатор з трьох шарів – 512 нейронів на першому, 128 на другому і 1 на останньому (оскільки проводиться вирішення задачі бінарної класифікації). Хід навчання протікав впродовж 15 епох. Функція активації на перших 2 шарах класифікатора – ReLU, на останньому – sigmoid. Оптимізатор – Rmsprop, функція втрат – бінарна крос-ентропія. Між повнозв'язними шарами також було введено шар регуляризації Dropout, що вимикає нейрони із ймовірністю 20%. Для додаткового навчання цих мереж так само застосовувалося доповнення даних.

Порівняння ступеню точності роботи і рівня помилки у тестових даних донавчених мереж із авторською CNN на основі U-Net, що була розроблена у ході виконання даної роботи, наведено у таблиці 3.2.

**Порівняння точності різних архітектур**

| Архітектура мережі | Точність (accuracy), % | Помилка (loss) |
|--------------------|------------------------|----------------|
| VGG16              | 92.52                  | 0.5670         |
| VGG19              | 91.13                  | 0.7121         |
| Modified U-NET     | 93.43                  | 0.1987         |

Із таблиці видно, що авторська CNN демонструє вищий рівень точності і менший відсоток помилки. Такі показники можуть бути зумовлені тим фактом, що знімки із набору даних Chest CT-ray Images суттєво відрізняються від зображень з набору даних ImageNet. Тобто, згорткові частини заздалегідь навчених мереж виділяють ознаки, що підходять для знімків ImageNet, але не підходять для знімків із масиву даних Chest X-ray Images, що був застосований у даній роботі. Крім того, зі збільшенням числа прихованих шарів у попередньо навчених нейронних мережах падає рівень їх якості роботи при вирішенні даної задачі. Так найкращу ступінь точності і найнижчу помилку продемонструвала мережа VGG16 з 16 шарами. Її показники можуть бути порівняні із показниками розробленої у даній роботі моделі CNN. Тим часом, мережі VGG19 і Inception V3 з більшим числом шарів нейронів (19 і 159 відповідно) продемонстрували гірші результати. Така тенденція говорить про те, що задля вирішення даної задачі не потрібно використання великої кількості нейронних шарів у згортковій частині CNN.

З метою підвищення рівня точності системи можна збільшити навчальну вибірку, розширити діапазон наявних у цій виборці знімків та ввести у розроблену систему додаткові дані про хворих. Наприклад, інформацію про стать та вік пацієнта.

На процес класифікації готовою мережею одного знімку знадобиться менше 1 секунди. Це уможливорює отримання від неї відповіді фактично миттєво. Це означає, що поставлені вимоги до рівня швидкості відгуку мережі, як і до рівня точності, було виконано.

### **Висновки до розділу 3**

У цьому розділі було проведено моделювання системи, проаналізовано та обрано засоби для програмної реалізації системи. Для вирішення описаної у попередніх розділах задачі було обрано мову програмування Python. Вона є оптимальною у даній ситуації, оскільки широко застосовується для організації машинного навчання.

Досліджено бібліотеки Python для роботи з нейронними мережами. Перелік основних особливостей наведено у третьому розділі кваліфікаційної роботи у пункті вибору технологій. Більшість бібліотек мають повну та скорочену версії, що дозволяє випробувати бібліотеки та обрати ту, яка буде найкраще вирішувати поставлені задачі перед розробниками.

Обрано базові програмні засоби, що необхідні для розробки системи. В якості платформи для розробки було використано Python та U-Net. Дана платформа є безкоштовною у використанні для некомерційних проектів та має інструменти для безпечної інтеграції бібліотек для роботи з машинного навчання. Проаналізовано результати дослідження.

## ВИСНОВКИ

Кваліфікаційна робота пов'язана з дослідженням алгоритмів розпізнавання зображень та з використанням методів машинного навчання у застосунках. Технологія розпізнавання образів стрімко розвивається та вже стала однією з провідних технологій. Використання даної технології допомагає підвищити наочність ефективність медичних досліджень та зробити більш точним та швидким.

У ході виконання кваліфікаційної роботи було вирішено ряд завдань, що описані нижче.

Проведено дослідження актуальності проблеми. Кожен досліджений алгоритм має певні особливості та певні аспекти його використання. Алгоритми базуються на обробці пікселів зображення, тож необхідно, щоб якість зображення, яке надається на вхід системи повинно мати задовільне освітлення, розміри тощо. Алгоритми беруть у розрахунок певні характеристики пікселів та проводиться їх подальша обробка.

Досліджено бібліотеки Python для роботи з нейронними мережами. Перелік основних особливостей наведено у третьому розділі кваліфікаційної роботи у пункті вибору технологій. Більшість бібліотек мають повну та скорочену версії, що дозволяє випробувати бібліотеки та обрати ту, яка буде найкраще вирішувати поставлені задачі перед розробниками.

Обрано базові програмні засоби, що необхідні для розробки системи. В якості платформи для розробки було використано Python та U-Net. Дана платформа є безкоштовною у використанні для некомерційних проектів та має інструменти для безпечної інтеграції бібліотек для роботи з машинного навчання. Проаналізовано результати дослідження.

В методичній частині було розроблено конспект лекції та лабораторної роботи, наведено повний зміст робіт разом з завданнями для самостійного виконання.

У розділі про охорону праці було досліджено основні нормативні документи, визначено норми умов праці робітника під час пандемії.

В результаті виконання кваліфікаційної роботи було створено проект в якому використовуються розпізнавання рентгенівських та КТ-знімнім для діагностування COVID-19. В проекті було досліджено такі алгоритми обробки зображення: VGG16, VGG19 та U-Net. Кожен алгоритм має похибки, але саме авторський алгоритм дав найбільш точний результат. Результати роботи було проаналізовано та зроблено висновки щодо роботи системи.

Подальшим кроком розвитку проекту є покращення сегментації зображення за рахунок комбінування алгоритмів та зменшення часу, що витрачається на обробку зображення.

## СПИСОК ВИКОРИСТАНИХ 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-19) outbreak—an update on the status. *Mil Med Res.* 2020;7(1):1-10.
2. Worldometer. COVID-19 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-19). 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-19). *Indian J Pediatr.* 2020;87(4):281-286.
8. Kong W., Agarwal P.P. Chest imaging appearance of COVID-19 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-19 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. 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.
13. 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.
14. Apostolopoulos I.D., Mpesiana TA. COVID-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med. 2020;43(2): 635–640.
15. Hall L.O., Paul R., Goldgof D.B., Goldgof G.M. Finding COVID-19 from chest X-Rays using deep learning on a small dataset. arXiv preprint arXiv:2004.02060; 2020.
16. Zhang J., Xie Y., Li Y., Shen C., Xia Y.. COVID-19 screening on chest X-Ray images using deep learning based anomaly detection. arXiv preprint arXiv:2003.12338; 2020.
17. Ucar F., Korkmaz D. COVIDiagnosis-net: deep BayesSqueezeNet based diagnostic of the corona virus disease 2019 (COVID-19) from X-ray images. Med Hypotheses. 2020;140 (2020):1–12.
18. Khalifa N.E.M., Taha M.H.N., Hassanien A.E., Elghamrawy S.. Detection of corona virus (COVID-19) 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.

19. Salman F.M., Abu-Naser SS, Alajrami E, Abu-Nasser BS, Alashqar BA, COVID-19 Detection Using Artificial Intelligence; United States: The DSpace Institutional Digital Repository System; 2020.  
<http://dstore.alazhar.edu.ps/xmlui/handle/123456789/587>.
20. Sethy P.K., Behera S.K. Detection of corona virus disease (COVID19) based on deep features. Preprints, 2020030300; 2020.
21. Maghdid H.S, Asaad A.T., Ghafoor K.Z., Sadiq A.S., Khan M.K. Diagnosing COVID-19 pneumonia from X-Ray and CT images using deep learning and transfer learning algorithms. arXiv preprint arXiv:2004.00038; 2020.
22. Bassi P.R., Attux R. A deep convolutional neural network for COVID-19 detection using chest X-rays. arXiv preprint arXiv: 2005.01578; 2020.
23. Ozturk T., Talo M., Yildirim E.A., Baloglu U.B., Yildirim O., Acharya U.R. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput Biol Med.* 2020;121:103792.
24. Mei X., Lee H.C., Diao K.Y., et al. Artificial intelligence-enabled rapid diagnosis of patients with COVID-19. *Nat Med.* 2020;26 (8):1224-1228.
25. 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.
26. Toraman S., Alakus T.B., Türkoglu I. Convolutional CapsNet: a novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. *Chaos Solitons Fractals.* 2020;140:110122.
27. Shoeibi A., Khodatars M., Alizadehsani R., et al. Automated detection and forecasting of COVID-19 using deep learning techniques: a review. arXiv preprint arXiv:2007.10785; 2020.
28. I.U. Khan, N. Aslam, A deep-learning-based framework for automated diagnosis of COVID-19 using X-ray images, *Information* 11 (2020) 419.
29. A.S.B. Aman Jaiswal, Analysis of deep learning algorithms on COVID-19 radiography database, *Int. J. Adv. Sci. Technol.* 29 (2020) 1268–1275.



30. 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.
31. 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-19 pneumonia?, 2020, ArXiv preprint arXiv: 2003.13145.
32. Aslan M.F., et al. "CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection." *Applied Soft Computing* (2021): 106912.
33. S. Asif, Y. Wenhui, Automatic detection of COVID-19 using X-ray images with deep convolutional neural networks and machine learning, *medRxiv* (2020).
34. M. Toğaçar, B. Ergen, Z. Cömert, COVID-19 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.
35. F. Ucar, D. Korkmaz, COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnostic of the coronavirus disease 2019 (COVID-19) from X-ray images, *Med. Hypotheses* (2020) 109761.
36. T. Ozturk, M. Talu, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, *Comput. Biol. Med.* (2020) 103792.
37. A.I. Khan, J.L. Shah, M.M. Bhat, Coronet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images, *Comput. Methods Programs Biomed.* (2020) 105581.
38. A. Sharma, S. Rani, D. Gupta, Artificial intelligence-based classification of chest X-ray images into COVID-19 and other infectious diseases, *Int. J. Biomed. Imaging* 2020 (2020).

39. 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.
40. 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-19 Detection using Lung Computerized Tomography Scan Data, 2020.
41. G. Liu, J. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification, *Neurocomputing* 337 (2019) 325–338.
42. L. Wang, A. Wong, COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images 2020, ArXiv preprint arXiv:2003.09871.
43. A. Gupta, Anjum, S. Gupta, R. Katarya, InstaCovNet-19: A Deep learning classification model for the detection of COVID-19 patients using chest X-ray, *Appl. Soft Comput.* (2020) 106859.
44. 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).
45. P.K. Sethy, S.K. Behera, Detection of coronavirus disease (covid-19) based on deep features, *Preprints* 2020030300 (2020) 2020.
46. 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.
47. M. Rahimzadeh, A. Attar, A new modified deep convolutional neural network for detecting COVID-19 from X-ray images, 2020, ArXiv preprint arXiv:2004.08052.
48. 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.

49. 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.
50. M. Farooq, A. Hafeez, Covid-resnet: A deep learning framework for screening of covid19 from radiographs, 2020, ArXiv preprint arXiv:2003.14395.
51. A.M. Dawud, K. Yurtkan, H. Oztoprak, Application of deep learning in neuroradiology: Brain haemorrhage classification using transfer learning, Comput. Intell. Neurosci. 2019 (2019).
52. D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, ArXiv preprint arXiv:1412.6980.
53. Tiwari S. A blur classification approach using deep convolution neural network. Int J Inf Syst Model Des. 2020;11(1):93-111.
54. Hinton G.E., Krizhevsky A., Wang S.D. Transforming autoencoders. International Conference on Artificial Neural Networks. Berlin, Heidelberg: Springer; 2011:44-51.
55. Sabour S., Frosst N., Hinton G.E. Dynamic routing between capsules. Adv Neural Inf Proces Syst. 2017;10(3):3856-3866.
56. 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>.
57. 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.
58. Dadario A.M.V. COVID-19 X rays. Kaggle. <https://doi.org/10.34740/KAGGLE/DSV/1019469>.
59. 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.

60. Дії роботодавця, якщо у працівника виявлено коронавірус.

<https://oppb.com.ua/news/diyi-robotodavcya-yakshcho-u-pracivnyka-vyyavleno-koronavirus>.

61. Постанови Кабінету міністрів України від 22.07.2020 р. No 641.

<https://zakon.rada.gov.ua/laws/show/641-2020-%D0%BF#Text>.

62. Законом України Про загальнообов'язкове державне соціальне страхування від 23.09.1999 р. No 1105-XIV.

<https://zakon.rada.gov.ua/laws/show/1105-14#Text>.

63. Левченко, Олег Григорович, et al. "Безпека життєдіяльності та цивільний захист." (2019).

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