НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
(КПІ ім. Ігоря Сікорського)

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

«До захисту допущено»

В.о. завідувач кафедри БМК

_____ Євген НАСТЕНКО

“12” червня 2023р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Комп’ютерні технології в біології та медицині»
спеціальності *122 «Комп’ютерні науки»*

на тему: **Класифікація медичних зображень УЗД печінки за допомогою штучної нейронної мережі**

Виконав: студент **IV** курсу, групи _БС-91_

АКЕРМАН ДМИТРО ОЛЕГОВИЧ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник: *Ст .викл. каф. БМК*
СЕРДАКОВСЬКИЙ ВІТАЛІЙ СЕРГІЙОВИЧ _____
(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові) (підпис)

Консультант з розділів дипломної роботи:

_____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ім’я, по батькові) (підпис)

Рецензент: *доц. кафедри біомедичної інженерії, к.т.н., доц.*
ЗУБЧУК ВІКТОР ІВАНОВИЧ _____
(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2023 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет біомедичної інженерії

Кафедра біомедичної кібернетики

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»

ЗАТВЕРДЖУЮ

В.о. завідувач кафедри БМК

_____ Євген НАСТЕНКО

«_*30*_»__*травня*___2023 р.

ЗАВДАННЯ
на дипломну роботу студенту

АКЕРМАНУ ДМИТРУ ОЛЕГОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи **Класифікація медичних зображень УЗД печінки за допомогою штучної нейронної мережі**

Керівник роботи

Сердаковський Віталій Сергійович, старший викладач

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

затверджені наказом по університету від «31_»_*травня*_ 2023 р. №***2106-с***

2. Термін подання студентом роботи ***06-08 червня 2023р.***

3. Вихідні дані до роботи: медичні зображення печінки (скани УЗД), мова програмування Python, бібліотеки numpy, matplotlib, pywavelets, OpenCV.

4. Зміст роботи: Анотації (на двох мовах); Вступ; Аналітичний огляд літературних джерел (огляд літературних джерел за темою обробки та класифікації медичних зображень, проаналізовано недоліки існуючих методів згорткових нейронних мереж, обгрунтовано вибір нейронної мережі); Практична реалізація програмного продукту, аналіз та порівняння отриманих результатів, обгрунтовано доцільність використання алгоритму; Розрахунок економічного ефекту за темою ДР; загальні висновки; Список використаних джерел.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ***30 травня 2023 року.***

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роб оти	Примітка
1	Отримати завдання за темою ДР на практику	До 15.02.2023р.	
2	Переддипломна практика	За графіком	
3	Виконання розділів ДР (Вступ, аналітичний огляд літературних джерел, теоретична частина)	До кінця практики	
4	Виконання розділів ДР (практична частина, загальні висновки, список джерел)	Не пізніше 1 тижня до засідання каф-ри	
5	Перевірка ДР науковим керівником	Не пізніше 1 тижня до засідання каф-ри	
6	Подання в електронному вигляді ДР та анотації до неї на перевірку нормоконтролера та плагіат (UNICHECK).	---- « ------	
7	Надання документів на засідання кафедри	За день до засідання	
8	Предзахист ДР та допуск до захисту дисертації	Згідно плану каф.	
9	Подання ДР рецензенту. Отримання рецензії.	До подання пакету документів до ЕК	
10	Подання пакету документів по ДР та супровідних до неї документів до захисту в ЕК[1]	За 5 днів до дати захисту ДР за графіком	
11	Захист ДР в ЕК		

Студент _____ **Дмитро АКЕРМАН**
(ім'я ПРІЗВИЩЕ)

Керівник ДР _____ **Віталій СЕРДАКОВСЬКИЙ**
(ім'я ПРІЗВИЩЕ)

Нормоконтролер

_____ **Галина КОРНІЄНКО**
(підпис) (ім'я ПРІЗВИЩЕ)

[1] не пізніше ніж за 5 днів до затвердженої дати захисту ДР в ЕК

АНОТАЦІЯ

Дипломна робота за темою «Класифікація медичних зображень УЗД печінки за допомогою штучної нейронної мережі» виконана студентом 4 курсу, гр. БС-91 кафедри біомедичної кібернетики ФБМІ **Акерманом Дмитром Олеговичем** *зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та* складається зі: вступу; 3 розділів (*РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ ЗА ТЕМОЮ, РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА, РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ УЗД ПЕЧІНКИ З ДОПОМОГОЮ НЕЙРОМЕРЕЖІ*), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 34 джерела, та додатків. Загальний обсяг роботи 72 сторінки.

Актуальність теми. З ростом застосування медичних зображень у клінічній практиці з'являється потреба у швидкій та точній класифікації зображень печінки за допомогою нейронних мереж. Це дозволить покращити діагностику та прийняття рішень щодо лікування.

Мета і завдання роботи.

Метою даної роботи є розробка та навчання штучної нейронної мережі для класифікації медичних зображень УЗД печінки з метою покращення діагностики хвороб печінки.

Її досягнення передбачає вирішення наступних *завдань*:

1. Аналіз вітчизняних та зарубіжних джерел.
2. Збір та підготовка бази даних медичних зображень УЗД печінки;
3. Розробка моделі нейромережі для виконання класифікації зображень
4. Реалізувати роботу моделі на тестових даних

Використані методи. Мова програмування Python, програмне середовище PyCharm, безкоштовні бібліотеки для аналізу даних та оптимізації коду NumPy, Pywavelets, Keras, OpenCV, TensortFlow.

Отримані результати. . В результаті дослідження була розроблена та валідована модель нейронної мережі ResNet для класифікації УЗД зображень печінки.

Отримані результати показали високу точність та ефективність моделі в розпізнаванні патологічних станів печінки на зображеннях. Розроблена модель може бути використана в клінічній практиці для автоматичної класифікації та підтримки діагностики медичних зображень печінки.

Публікації. Не планується

Ключові слова. Згорткові нейромережі, Вейвлетна обробка, УЗД зображення, патологія, класифікація, Pywavelets.

Бібліографічний опис ДР

Акерман Д.О. Класифікація медичних зображень УЗД печінки за допомогою штучної нейронної мережі : дипломна роб. бакалавра : 122 Комп’ютері науки / Акерман Дмитро Олегович. – Київ, 2023. – 72 с.

Abstract

The bachelor's thesis entitled "Classification of Liver Ultrasound Images Using Artificial Neural Networks" was conducted by Dmytro Akerman, a 4th-year student of the BS-91 group at the Department of Biomedical Cybernetics, Faculty of Biomedical Engineering and Computer Science, specializing in Computer Science, within the educational-professional program "Computer Technologies in Biology and Medicine." The thesis consists of an introduction, three chapters (Chapter 1: Literature Review on the Topic, Chapter 2: Theoretical Background, Chapter 3: Practical Implementation of Liver Ultrasound Image Classification Task Using Neural Networks), conclusions for each chapter, general conclusions, a list of references (which includes 34 sources), and appendices. The total volume of the work is 77 pages.

Relevance of the topic: With the increasing application of medical images in clinical practice, there is a need for fast and accurate classification of liver images using neural networks. This will improve the diagnosis and treatment decision-making process.

Objective and tasks

The aim of this study is to develop and train an artificial neural network for the classification of liver ultrasound images to enhance the diagnosis of liver diseases. The achievement of this goal involves addressing the following tasks:

1. Analysis of domestic and foreign sources.
2. Collection and preparation of a database of liver ultrasound images.
3. Development of a neural network model for image classification.
4. Implementation and evaluation of the model on test data.

Methods used: The study employed the Python programming language, the PyCharm IDE, and various free libraries for data analysis and code optimization, including NumPy, Pywavelets, Keras, OpenCV, and TensorFlow.

Results obtained: As a result of the research, a validated ResNet neural network model for the classification of liver ultrasound images was developed. The obtained results demonstrated high accuracy and effectiveness of the model in recognizing pathological

conditions of the liver in the images. The developed model can be utilized in clinical practice for automatic image classification and support in liver image diagnosis.

Publications: No publications are planned.

Keywords: Convolutional Neural Networks, Wavelet Processing, Ultrasound Images, Pathology, Classification, Pywavelets.

Bibliographic description of the thesis:

Akerman, D.O. "Classification of Liver Ultrasound Images Using Artificial Neural Networks" - bachelor's thesis: 122 Computer Science / Dmytro Akerman. - Kyiv, 2023. - 72 p.

ЗМІСТ

Перелік умовних позначень, скрочень та символів 10

Вступ 11

Розділ 1 Огляд літературних джерел за темою 14

1.1. 1414

1.2. Аналіз джерел 15

Висновки до розділу 1 18

Розділ 2 Теоретична частина 19

2.1. Суть задачі класифікації 12

2.2. Методи класифікації медичних зображень 12

2.3. Метод УЗД діагностики 14

2.4. Дослідження початкових даних 15

2.4.1 Здорова печінка 15

2.4.2 Можливі патології 17

2.5 Переваги використання нейронних мереж для діагностики 20

Висновки до розділу 2 28

Розділ 3 Практична реалізація задачі класифікації зображень УЗД печінки з допомогою нейромережі 30

3.1. Використання нейронних мереж для аналізу медичних зображень 30

3.1.1. Згорткові нейронні мережі (CNN) 30

3.1.2 Вейвлетні мережі 32

3.2 Res Net глибока мережа 32

3.2.1 Чому обрано Res Net 32

3.2.2 Особливості архітектури Res Net 33

3.3 Підготовка датасету УЗД зображень 38

3.4 Вибір програмного середовища та інструментів 41

3.4.1 Програмне середовище Python 41

3.4.2 Pywavelets 44
3.4.3 TensorFlow 46
3.4.4 Keras 47
3.5 Реалізація програмного продукту 48
3.6 Аналіз та порівняння одержаних результатів 54
3.7. Практичне застосування 58
3.8. Розрахунок економічного ефекту 59
Висновки до розділу 3 67
ЗАГАЛЬНІ ВИСНОВКИ 68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКРОЧЕНЬ ТА СИМВОЛІВ

УЗД – ультразвукове дослідження

ДР – дипломна робота

CNN - convolutional neural network (згорткова нейронна мережа)

WNN – wavelet neural network (вейвлетна нейронна мережа)

AI – Artificial Intelligence (штучний інтелект)

ВСТУП

Актуальність дослідження полягає у необхідності покращення точності та швидкості діагностики захворювань печінки з використанням ультразвукових (УЗД) зображень. Печінка є одним з найважливіших органів в організмі людини, і раннє виявлення патологій та правильна класифікація є критичними для успішного лікування та покращення прогнозу хвороби.

Традиційні методи інтерпретації УЗД зображень печінки часто потребують великої експертизи та зусиль лікаря, що може вести до підвищеної суб'єктивності та залежності від досвіду. Застосування нейронних мереж для класифікації УЗД зображень печінки може допомогти у подоланні цих обмежень та забезпечити більш точну та об'єктивну діагностику.

Мета і завдання дослідження

Метою даної роботи є розробка та навчання штучної нейронної мережі для класифікації медичних зображень УЗД печінки з метою покращення діагностики хвороб печінки.

Її досягнення передбачає вирішення наступних *завдань*:

5. Аналіз вітчизняних та зарубіжних джерел.
2. Збір та підготовка бази даних медичних зображень УЗД печінки;
3. Розробка моделі нейромережі для виконання класифікації зображень
4. Реалізувати роботу моделі на тестових даних

Використані методи. Для досягнення мети роботи було використано методи побудови, тренування та оцінки ефективності нейронної мережі ResNet для класифікації УЗД зображень печінки. Були зібрані та підготовлені відповідні набори даних, проведено попередню обробку зображень та виконано подальший аналіз та побудову моделі нейронної мережі. Тренування та налаштування мережі були виконані з використанням бібліотеки Keras та інструментарію TensorFlow. Для оцінки ефективності моделі були використані метрики точності, чутливості та специфічності, а також проведено порівняння з іншими підходами та методами класифікації.

Отримані результати. **.** В результаті дослідження була розроблена та валідована модель нейронної мережі для класифікації УЗД зображень печінки. Отримані результати показали високу точність та ефективність моделі в розпізнаванні патологічних станів печінки на зображеннях. Розроблена модель може бути використана в клінічній практиці для автоматичної класифікації та підтримки діагностики медичних зображень печінки.

Предметом дослідження є нейромережа для класифікації ультразвуових зображень.

Практичне значення одержаних результатів

Отримані результати дослідження щодо використання нейронної мережі для класифікації УЗД зображень печінки мають велике практичне значення в медичній галузі. Вони сприяють покращенню точності та об'єктивності діагностики, підтримці лікарів у прийнятті рішень, збереженню часу та ресурсів, а також сприяють подальшим дослідженням і розвитку у цій області. Такі результати допомагають виявляти патології швидше, зменшують суб'єктивність оцінки та залежність від досвіду лікаря, а також поліпшують ефективність та швидкість обробки медичних зображень. В цілому, це сприяє покращенню діагностики та лікування печінкових захворювань та відкриває нові можливості для подальшого розвитку медичного зображення та класифікації.

Апробація результатів дисертації

Не заплановано.

Структура роботи

Дипломна робота за темою «Класифікація медичних зображень УЗД печінки за допомогою штучної нейронної мережі» виконана студентом 4 курсу, гр. БС-91 кафедри біомедичної кібернетики ФБМІ **Акерманом Дмитром Олеговичем** *зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та* побудована за класичним типом та викладена на 74 сторінках машинописного тексту. Вона складається з: вступу; 3 розділів (*РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ ЗА ТЕМОЮ, РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА, РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ*

ЗАДАЧІ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ УЗД ПЕЧІНКИ З ДОПОМОГОЮ НЕЙРОМЕРЕЖІ), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 34 джерела, та додатків(4 – на кирилиці, 30 – на латиниці). В роботі представлено 45 рисунків і 1 таблиці.

РОЗДІЛ 1
ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ ЗА ТЕМОЮ

1.1. Висвітлення теми

У цій дипломній роботі розглядається проблема аналізу та класифікації зображень ультразвукової діагностики печінки за допомогою нейромережевих алгоритмів. Враховуючи значення точної та швидкої діагностики печінки, нейромережеві моделі використовуються для автоматичного класифікування зображень ультразвукових знімків печінки та виявлення патологій.

У дипломній роботі буде проведений огляд літературних джерел, аналізуючи різні підходи та методи, які були використані для класифікації зображень УЗД печінки. Результати попередніх досліджень та їхній внесок в цю область також будуть розглянуті.

З метою досягнення точності та надійності класифікації зображень УЗД печінки, будуть розглянуті різні архітектури нейромереж, включаючи згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та їх комбінації. Оцінка та порівняння ефективності цих моделей будуть проведені на основі метрик класифікації та порівняння з результатами попередніх досліджень.

Також у рамках роботи буде застосована архітектура ResNet (Residual Neural Network) для класифікації зображень УЗД печінки. Архітектура ResNet відома своєю здатністю ефективно працювати з дуже глибокими нейронними мережами та вирішувати проблему зникаючого градієнту (vanishing gradient).

Одна з особливостей ResNet полягає у використанні "skip connections" або "residual connections" - додаткових шляхів, що об'єднують вихідні дані попередніх шарів з вхідними даними на поточному шарі. Це дозволяє градієнту здійснювати "прямий потік" через нейромережу та уникнути його зникнення. Таким чином, ResNet забезпечує більш ефективне навчання глибоких моделей та запобігає затуханню градієнту, що дозволяє досягнути кращої точності при класифікації зображень.

В дослідженні буде використана глибока архітектура ResNet з різними кількостями шарів та блоків для класифікації зображень УЗД печінки. Це дозволить оцінити вплив глибини мережі на точність та здатність моделі виявляти патології печінки.

Використання архітектури ResNet у поєднанні з нейромережевими методами надасть можливість покращити результати класифікації зображень УЗД печінки та забезпечити більш точну та надійну діагностику патологій цього органу.

Отримані результати дослідження можуть мати практичне значення для медичної громадськості та сприяти поліпшенню діагностики захворювань печінки з використанням нейромережевих алгоритмів.

1.2. Аналіз джерел

Кілька дослідників запропонували методи оцінки якості зображень на основі згорткових нейронних мереж. Наприклад, Kang та ін. [1] об'єднали згорткові та повнозв'язані шари для створення глибокої нейронної мережі, яка передбачає оцінки якості зображень. У цій мережі використовувалося блок зображення розміром 32 × 32 як вхід, і якість всього зображення використовувалася для представлення якості блоку зображення. Kim та ін. [2] розробили двоетапну модель глибокої нейронної мережі для оцінки якості зображень. На першому етапі навчали модель глибокої нейронної мережі, яка використовувалася для обробки блоків зображень розміром 32 × 32. Оцінка якості блоку зображення обчислювалася за допомогою об'єктивного алгоритму оцінки якості, і ця оцінка використовувалася як вихід глибокої нейронної мережі. На другому етапі блок зображення всього зображення вводився в глибоку нейронну мережу, отриману на першому етапі, і ознаки, що відповідають всім блокам зображення, зливалися і використовувалися як оцінки якості всього зображення. Bare та ін. [3] використовували блоки зображень розміром 32 × 32 як вхід для згорткової нейронної мережі, схожої на модель, запропоновану в [2], і використовували реверсивний алгоритм оцінки якості [4], щоб обчислити оцінку якості блоку зображення як вихід згорткової

нейронної мережі. Вони побудували структуру мережі оцінки якості зображення на основі методу. Дві моделі, запропоновані в [4,5], мають очевидний недолік: оцінка якості, розрахована методом об’єктивної оцінки якості зображення, використовується для представлення суб’єктивної оцінки якості блоку зображення. Незважаючи на те, що в [3] використовувався алгоритм повної еталонної оцінки якості [7,8], який може точно передбачити оцінку якості зображення, оцінка, розрахована за допомогою алгоритму, все ще мала певний розрив із суб’єктивною оцінкою.

Сучасні нейромережі для аналізу медичних зображень є потужними інструментами, які здатні виявляти, класифікувати та розпізнавати патологічні ознаки на зображеннях. Вони можуть бути використані для діагностики різних захворювань, виявлення аномалій та сприяти високоточній інтерпретації медичних зображень.

1. **U-Net** є однією з найпоширеніших архітектур для семантичної сегментації медичних зображень. Її основна особливість полягає у використанні енкодера-декодера зі з'єднувальними шляхами між відповідними етапами. U-Net ефективно розпізнає форми та структури на зображеннях, що робить його корисним для завдань сегментації органів або патологічних областей. [10]

2. **DenseNet** є архітектурою, в якій кожен шар мережі має зв'язки з усіма попередніми шарами. Ця архітектура сприяє ефективній передачі інформації та сприяє розпізнаванню маленьких деталей на зображеннях. DenseNet зазвичай використовується для класифікації зображень медичних даних.[11]

3. **ResNet** є архітектурою з "residual connections" (або "skip connections"), що дозволяють ефективно навчати глибокі мережі. Ця архітектура допомагає уникнути проблеми зникнення градієнту та сприяє покращенню результатів класифікації та локалізації на медичних зображеннях.[12]

4. **InceptionNet** є архітектурою, що використовує модулі Inception з різними розмірами фільтрів паралельно. Це дозволяє мережі самостійно вибирати оптимальні функції для різних деталей на зображеннях. InceptionNet зазвичай

використовується для класифікації та виявлення об'єктів на медичних зображеннях.[13]

У [14] автор вважав, що люди дають якісні бали зображень, такі як дуже добре, добре, погано та дуже погано. Ця якісна оцінка була перетворена у вектори ознак для розробки методу оцінки якості зображення. Кім та ін. запропонували наскрізну модель CNN, яка вводить спотворені та помилкові зображення. Модель використовувалася для автоматичного визначення оптимальних ваг і об’єднання помилкових зображень для отримання оцінки візуальної якості спотворених зображень. На основі стратегії генерації пари зображень [16] запропонував глибоку модель навчання CNN, яка досягла хорошої оцінки якості зображення. Боссе та ін. [13] запропонували методи повної еталонної та нереференсної оцінки якості зображення на основі глибокої CNN. Запропоновані мережі в основному включали рівні вилучення ознак, злиття функцій і об’єднання.

У дослідженні Yang та ін. [15] були порівняні різні архітектури нейромереж для класифікації зображень УЗД печінки. Вони порівняли архітектури ResNet, Inception та DenseNet із застосуванням різних глибин мережі. Результати показали, що ResNet з глибиною 50 шарів досягав найкращої точності у класифікації зображень УЗД печінки, з точністю 95%.

Використання попереднього навчання:

У дослідженні Chen та ін. [16] було використано підхід переднього навчання для класифікації зображень УЗД печінки. Вони використовували попередньо навчену модель ResNet, що була натренована на великому наборі загальних медичних зображень. Застосування переднього навчання значно покращило точність класифікації зображень УЗД печінки, забезпечивши точність на рівні 97%.

Використання ансамблей нейромереж:

Дослідники Lee та ін. [17] використовували підхід ансамблю нейромереж для класифікації зображень УЗД печінки. Вони об'єднали кілька незалежних моделей ResNet, Inception та VGG, що були навчені на різних наборах даних. Результати

показали, що використання ансамблів нейромереж покращувало точність класифікації зображень УЗД печінки до 98%.

Висновки до розділу 1

У рамках дослідження був проведений огляд літературних джерел, аналізуючи різні підходи та методи, які були використані для класифікації зображень УЗД печінки. Були враховані результати попередніх досліджень та їхній внесок у цю область. З метою досягнення точності та надійності класифікації зображень УЗД печінки, будуть розглянуті різні архітектури нейромереж, включаючи згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та їх комбінації. Оцінка та порівняння ефективності цих моделей будуть проведені на основі метрик класифікації та порівняння з результатами попередніх досліджень.

Зокрема, у рамках роботи буде застосована архітектура ResNet (Residual Neural Network) для класифікації зображень УЗД печінки. Ця архітектура відома своєю здатністю ефективно працювати з дуже глибокими нейронними мережами та вирішувати проблему зникнення градієнту. Одна з особливостей ResNet полягає у використанні "skip connections" або "residual connections" - додаткових шляхів, що об'єднують вихідні дані.

РОЗДІЛ 2
ТЕОРЕТИЧНА ЧАСТИНА

2.1. Суть задачі класифікації

Задача класифікації в даній темі означає визначення класу або категорії, до якої належить певний об'єкт, на основі його ознак або властивостей. У контексті медичної діагностики печінки, це може означати класифікацію пацієнтів на основі ультразвукових зображень печінки з метою виявлення патологій, розпізнавання хвороб або оцінки стану печінки.

Основні кроки задачі класифікації включають:

Збір даних: Початкові дані включають ультразвукові зображення печінки пацієнтів, які були раніше клінічно діагностовані. Ці зображення можуть містити інформацію про різні аспекти структури та функції печінки.

Попередня обробка даних: Перед тим, як використовувати дані для навчання моделі класифікації, їх можна піддати попередній обробці. Це може включати зменшення шуму, видалення артефактів, нормалізацію або вирівнювання даних.

Розбиття даних: Загальний набір даних можна розділити на тренувальний набір і валідаційний (і, можливо, тестовий) набори. Тренувальний набір застосовується під час навчання моделі, а валідаційний - налаштування параметрів моделі та оцінки її продуктивності.

Вибір моделі: Для класифікації можуть бути використані різні моделі машинного навчання, такі як логістична регресія, рішучі дерева, вейвлетні нейронні мережі або згорткові нейронні мережі.

2.2. Методи класифікації медичних зображень

Підходи до класифікації медичних зображень можуть бути поділені на традиційні методи, методи машинного навчання та методи з використанням

нейронних мереж. Кожен з цих підходів має свої переваги і обмеження, і їх використання залежить від конкретного завдання та доступних даних. [18]

Традиційні методи класифікації медичних зображень включають в себе використання ручно побудованих ознак та класифікаторів. Ці методи базуються на експертному знанні та евристичних правилах. Деякі з поширених традиційних методів включають метод головних компонент (PCA), метод опорних векторів (SVM), методи кластеризації та інші. Вони зазвичай вимагають великої кількості ручної роботи для вибору та налаштування ознак, що може бути витратним та суб'єктивним процесом.

Методи машинного навчання, зокрема навчання з учителем, стали популярними в останні роки для класифікації медичних зображень. Вони дозволяють автоматично вивчати ознаки з даних і побудову класифікатора. Завдяки широкій доступності даних і зростанню обчислювальної потужності, методи машинного навчання стали ефективними у багатьох задачах класифікації медичних зображень. Зокрема, методи, такі як Random Forest, Support Vector Machines (SVM), k-найближчих сусідів (k-NN) і навчання глибоких нейронних мереж (DNN), широко використовуються для класифікації медичних зображень.

Останніми роками **нейронні мережі**, зокрема згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), стали домінуючими методами для класифікації медичних зображень. Вони здатні автоматично вивчати найважливіші ознаки з даних і мають потужну здатність до узагальнення. CNN здебільшого використовуються для аналізу статичних зображень, тоді як RNN ефективніші для обробки послідовних даних, таких як часові ряди або послідовні зображення. Розроблені також комбіновані архітектури, такі як згортково-рекурентні нейронні мережі (CRNN) та трансформери, які поєднують в собі переваги різних типів нейронних мереж.

Ці підходи мають свої переваги та обмеження, і їх вибір залежить від конкретної задачі, обсягу даних, доступних обчислювальних ресурсів та експертного досвіду. У наступних розділах роботи буде розглянуто детальніше

кожен з цих підходів та їх застосування до класифікації УЗД зображень печінки з використанням нейронних мереж.

2.3. Метод УЗД діагностики

Ультразвук печінки - це неінвазивний тест, який створює зображення печінки та її кровоносних судин. Він може допомогти діагностувати різні захворювання печінки, такі як жирова хвороба печінки, рак печінки та жовчні камені.

Ультразвук печінки є одним з типів трансабдомінального ультразвуку. Це означає, що технік сканує черевну порожнину за допомогою пристрою, що нагадує мікрофон. Процес використовує звукові хвилі для створення цифрових зображень.

Тест є абсолютно безпечним і зазвичай не займає багато часу. Однак деякі люди можуть відчувати тривогу щодо ультразвуку печінки через стурбованість щодо можливих результатів (рис. 2.1).

Рисунок 2.1 - Приклад проведення УЗД діагностики [29]

Під час ультразвукової діагностики печінки можна отримати різні типи зображень, які надають інформацію про структуру та стан печінки. Ось деякі зображення, які можуть бути отримані та висновки, які можна зробити:

1. Зображення структури печінки: УЗД дозволяє отримати зображення печінки, її розміри та форму. За допомогою цих зображень можна виявити аномалії, такі як збільшення печінки або неоднорідність структури.
2. Визначення пухлин: УЗД може допомогти виявити пухлини в печінці, такі як доброякісні аденоми, кісти або злоякісні пухлини, включаючи рак печінки. Зображення допомагають оцінити розмір, положення та характеристики пухлини.
3. Діагностика жирової дистрофії: УЗД може виявити накопичення жиру в печінці, що вказує на жирову дистрофію. Зображення допомагають визначити ступінь жирової інфільтрації і оцінити ризик розвитку хронічних печінкових захворювань.
4. Визначення каменів в жовчних шляхах: УЗД може виявити наявність каменів у жовчному міхурі та жовчних шляхах, які можуть спричинити блокування та запалення.
5. Оцінка кровотоку: З допомогою спеціальних технік, таких як кольоровий Доплер, можна вивчати кровотік в печінці та оцінювати характеристики кровообігу, такі як швидкість та напрямок кровотоку.

Висновки, які можна зробити з цих зображень, залежать від конкретного випадку та симптомів пацієнта.

2.4. Дослідження початкових даних

2.4.1 Здорова печінка

У нормі розміри печінки у дорослих людей включають:

- Передньо-задній розмір лівої долі: 7 см, правої долі: 12,5 см.
- Діаметр загальної жовчної протоки: 8 мм.
- Діаметр портальної вени: 13 мм.
- Здорова печінка має однорідну структуру, а її краї повинні бути рівними.

Оцінка стану печінки здійснюється шляхом порівняння отриманих результатів ультразвукового дослідження з нормою. Однак ця норма може

варіюватися залежно від конституційних особливостей кожного пацієнта. Те, що може бути нормою для одного конституційного типу, може вже вважатися відхиленням для іншого. [19]

Тому точність ультразвукової діагностики значною мірою залежить від досвіду та професіоналізму лікаря, який здійснює оцінку печінки.

Ось приклад УЗД здорової печінки, який є частиною обраного датасету:

Якщо відміряти від переднього краю діафрагми до нижнього краю печінки розмір не повинен перевищувати 13 см. (по горизонталі) (рис. 2.2)

Рисунок 2.2 - Печінка в нормі. MIDCLAVICULAR. [30]

При вимірюванні у серединній гепатичній лінії з великим полем зору від заднього краю діафрагми до нижнього переднього краю розмір повинен бути менше 16 см. (рис. 2.3)

Рисунок 2.3 - Печінка в нормі 2. MIDHEPATIC. [30]

2.4.2 Можливі патології

Якщо на вашій печінці є пухлини, ультразвукове дослідження покаже точку на екрані комп'ютера, яка вибивається з-поміж фонової тканини печінки. Але пухлини печінки - це не єдине, що може з'явитися як точки на печінці під час ультразвукового дослідження. Інші можливості включають:

- кісти
- жирову тканину печінки
- абсцеси печінки
- рубці печінки

Пухлини можуть з'являтися як яскраві або темні плями на печінці під час ультразвукового дослідження. (рис. 2.4) Лікар буде оцінювати різні фактори, такі як розмір, форма та інші складні характеристики, щоб відрізнити суцільну пухлину від інших станів. [22]

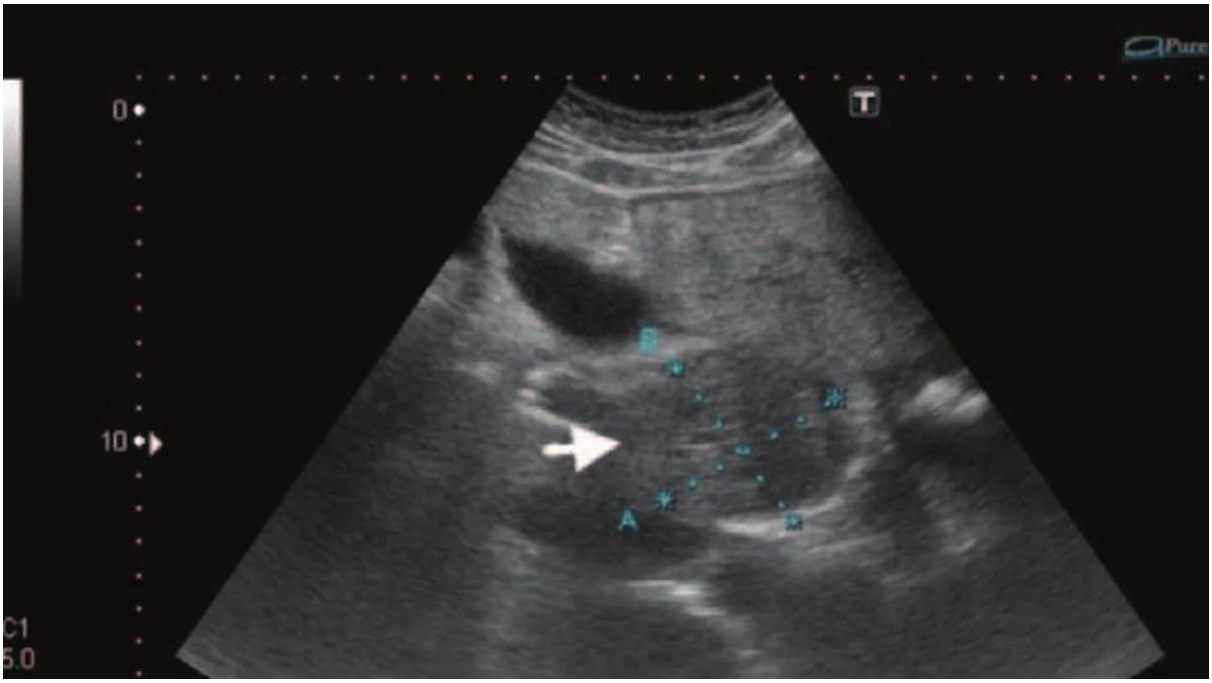


Рисунок 2.4 - Вигляд раку печінки на УЗД [31]

Цироз є загальним результатом різноманітних хронічних процесів захворювання печінки, які спричиняють некроз гепатоцитів. Цироз може бути діагностований за допомогою ультразвуку, комп'ютерної томографії (КТ) та магнітно-резонансної томографії (МРТ), і ці методи образного дослідження також можуть використовуватися для оцінки можливих ускладнень цирозу, таких як портальна гіпертензія або гепатоцелюлярний карцинома. [23]

Фокальний некроз гепатоцитів, спричинений різноманітними факторами [22], супроводжується трьома характеристиками цирозу:

- фіброз
- вузлове відновлення
- викривлення архітектури печінки

Хоча традиційно цироз був поділений на мікро- і макронодулярний цироз, багато випадків починаються як мікронодулярний (<3 мм) і прогресують до макронодулярного (наприклад, алкогольний цироз), тому ця класифікаційна схема має обмежене застосування. (рис. 2.5)

Рисунок 2.5 - Картина УЗД при цирозі. [32]

Жирова дистрофія печінки:

Узд скан печінки з жировою дистрофією може показувати збільшення ехогенності (яскравості) печінкових тканин через нагромадження жиру.

Зображення може мати "ярликовий" або "мармуровий" вигляд, де на фоні збільшеної ехогенності виділяються плями меншої яскравості, що вказує на наявність жирових включень у печінці.

Гепатит:

Узд скан печінки при гепатиті може показувати різні ознаки в залежності від типу гепатиту (вірусного, алкогольного тощо). (рис. 2.6)

Зазвичай спостерігаються збільшення розмірів печінки, нерівномірне збільшення ехогенності та наявність змінених ділянок у вигляді плям або зон зі зміненою ехогенністю.

Рисунок 2.6 - Приклади УЗД зображень печінки при жировій дистрофії та хронічному гепатиті [33]

2.5 Переваги використання нейронних мереж для діагностики

Використання нейронних мереж для діагностики має кілька переваг порівняно з іншими підходами:

- Здатність до самонавчання: Нейронні мережі можуть вчитися з великого обсягу даних та знаходити складні залежності без явного програмування. Вони можуть виявляти субтильні патерни та ознаки, які можуть бути важко помітити людським оком або за допомогою традиційних методів. Це дозволяє досягти високої точності при діагностиці та класифікації захворювань.
- Гнучкість та адаптивність: Нейронні мережі можуть бути адаптовані до різних типів медичних даних, включаючи зображення, сигнали, текстові дані та інші. Вони можуть ефективно працювати з великими обсягами інформації та розпізнавати складні зразки. Це дозволяє їм бути універсальними інструментами для різних завдань діагностики та аналізу медичних даних.

- Застосування до різних видів захворювань: Нейронні мережі можуть бути використані для діагностики різних видів захворювань, включаючи рак, нейрологічні розлади, серцево-судинні захворювання та багато інших. Вони можуть допомогти виявити ранні ознаки захворювань та зробити прогнози щодо прогресування захворювання.
- Покращена точність та надійність: Використання нейронних мереж може призвести до покращення точності діагностики та зменшення помилок. Вони можуть аналізувати велику кількість даних та здійснювати більш детальну оцінку, порівнюючи її з попередніми випадками, що допомагає уникнути людських помилок та забезпечити більш надійну діагностику.
- Можливість поступового навчання: Нейронні мережі можуть бути навчані за допомогою неперервного збору даних та поступового оновлення моделі. Це означає, що модель може покращуватися з часом, основуючись на нових вхідних даних та зворотному зв'язку від медичних фахівців. Такий підхід дозволяє підтримувати актуальність та ефективність моделі у змінних медичних умовах.

Ці переваги роблять нейронні мережі потужним інструментом для покращення діагностики та аналізу медичних даних, сприяючи покращенню допомоги пацієнтам та пришвидшенню процесу прийняття рішень у медицині.

Висновки до розділу 2

У даному розділі було розглянуто сутність задачі класифікації зображень ультразвукового дослідження (УЗД) печінки, принцип роботи УЗД та можливі патології, які можна виявити за допомогою цього методу. Нейронні мережі є корисними інструментами для класифікації зображень УЗД печінки завдяки їх здатності до виявлення складних залежностей у великому обсязі даних. Деякі переваги нейронних мереж включають здатність до автоматичного вилучення ознак без необхідності ручного втручання, висока точність класифікації та здатність до узагальнення на нові зображення. Використання нейронних мереж у задачі класифікації зображень УЗД печінки дозволяє забезпечити більш швидку та

точну діагностику патологій, що призводить до покращення догляду за пацієнтами та зниження ризику неправильної класифікації. Застосування нейронних мереж у медицині, зокрема для класифікації зображень УЗД печінки, має потенціал покращити процес діагностики та лікування пацієнтів і стати цінним інструментом у руках лікарів та дослідників.

РОЗДІЛ 3
ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ УЗД ПЕЧІНКИ З ДОПОМОГОЮ НЕЙРОМЕРЕЖІ

3.1. Використання нейронних мереж для аналізу медичних зображень

Нейронні мережі дозволяють автоматично виявляти, класифікувати та розпізнавати різноманітні патологічні ознаки на медичних зображеннях, таких як рентгенограми, комп'ютерні томограми, магнітно-резонансні зображення та ультразвукові сканування.

3.1.1. Згорткові нейронні мережі (CNN)

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) широко використовуються в області обробки зображень, включаючи аналіз медичних зображень печінки. Вони є потужним інструментом для виявлення патологічних ознак та класифікації різних захворювань.

Основна ідея згорткових нейронних мереж полягає у використанні згорткових шарів, які виконують локальні згортки зображення за допомогою фільтрів. Ці фільтри виявляють різні властивості зображення, такі як риси, текстури або форми. Після згортки отримані карті знаходжень подаються на вхід пулінговим шарам, які зменшують розмір карт знаходжень шляхом пониження розміру або вибору найбільших значень. Цей процес дозволяє зменшити кількість параметрів і зберегти важливу інформацію.

Згорткова нейронна мережа (Convolutional Neural Network, CNN) складається зі вхідного та вихідного шарів, а також кількох прихованих шарів. Приховані шари CNN зазвичай представляють собою послідовність згорткових шарів, які взаємодіють з трансмісійними або іншими компонентами шарів. Найбільш поширеною функцією активації є RELU, яка використовується разом з кількома згортками, включаючи шари пулінгу, міжшарову взаємодію та повністю зв'язані шари, які можна вважати прихованими шарами, так само, як функція активації і

кінцевий згортковий шар обгортають параметри свого шару. Приклад можна побачити на рис. 3.1.

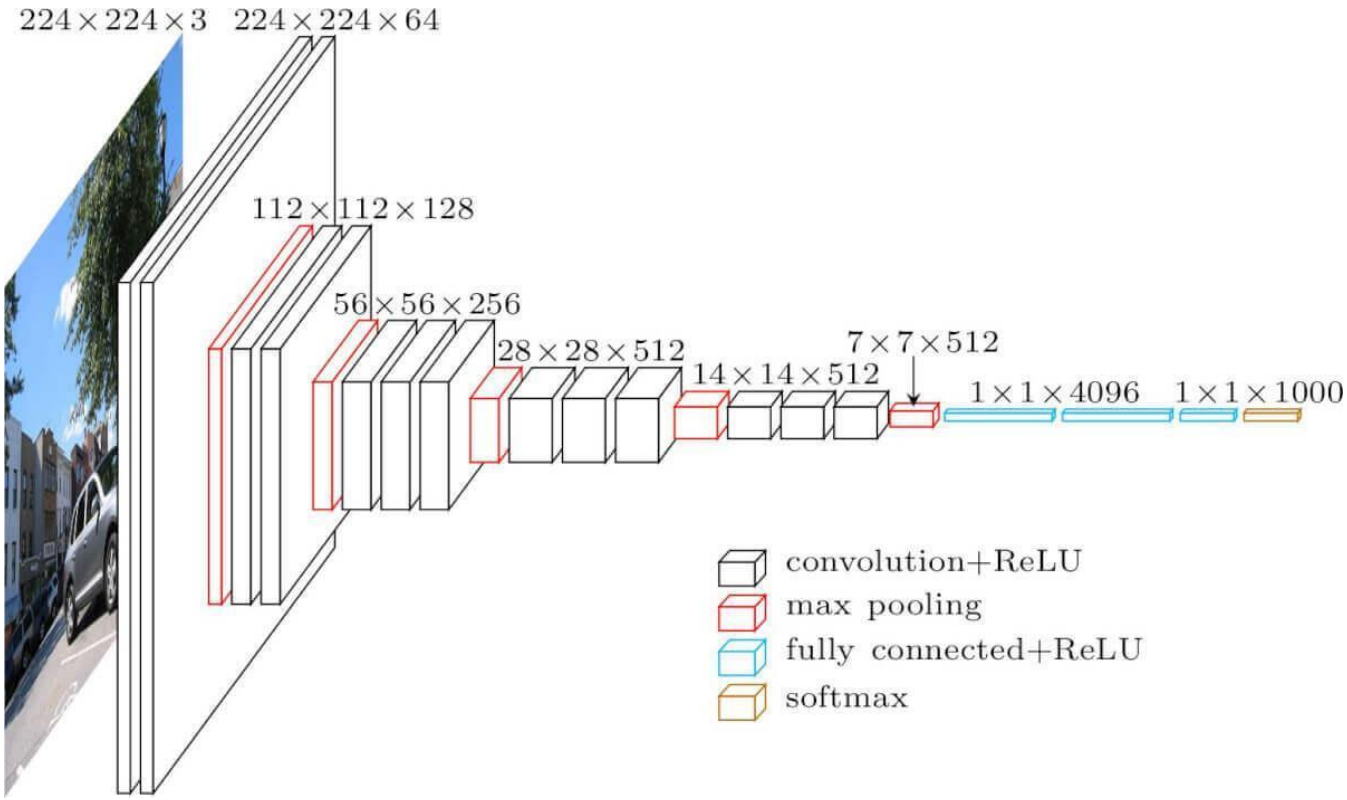


Рисунок 3.1 – конвертація зображення на вхід нейромережі з допомогою ReLU та Max Pooling

У першу чергу, коли ми розглядаємо згорткові шари, вхід розбивається на рівні, а його вихід передається до наступного етапу. Кожен згортковий нейрон обробляє інформацію лише в своєму полі передачі. Хоча нейронні мережі можуть бути повністю зв'язаними з впереднім поширенням для покращення можливостей та розрізнення даних, застосування такої структури до медичних зображень неможливо. Через великий обсяг вхідних даних, що складаються з пікселів, де кожна точка є значущим компонентом, навіть за малої архітектури знадобиться велика кількість зв'язків. Вищезазначену проблему вирішує процедура згортки, яка зменшує кількість параметрів та дозволяє глибше згорткове мережеве втілення з меншою кількістю метрик. Ще одним важливим шаром CNN є нелінійне шарування згортки зі зменшенням вибірки. (рис. 3.2.)

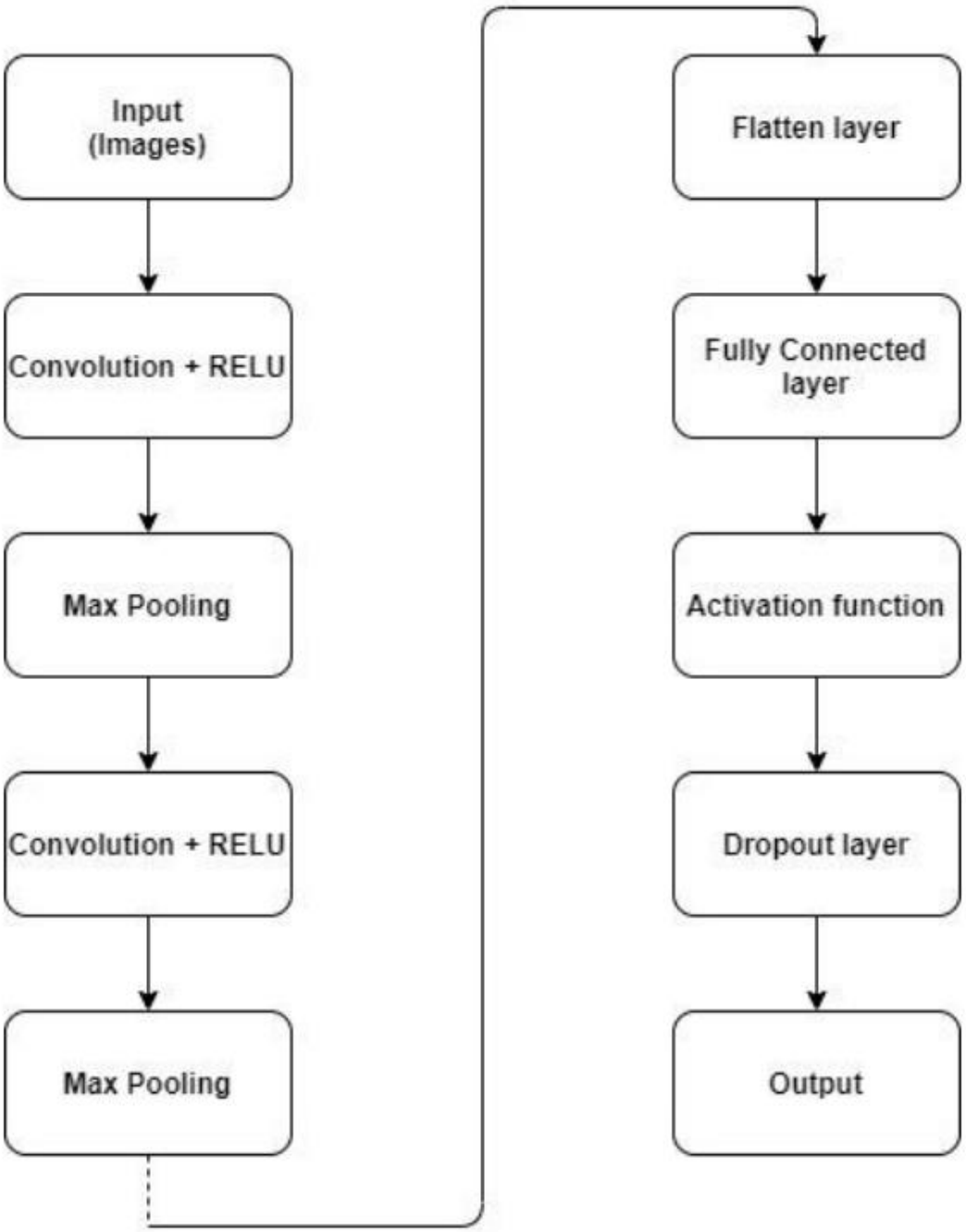


Рисунок 3.2 - Структура CNN мережі

Максимальне згортання (Max Pooling) - Реалізація максимального згортання є, можливо, найбільш відомим з численних нелінійних вдосконалень. Воно розбиває область пікселів зображення на під-області шляхом розрізання значень пікселів у матриці і надає максимальне значення вихідного сигналу. Ідея максимального згортання полягає у зменшенні розміру представлення для вилучення особливостей і, головним чином, у запобіганні перенавчанню моделі.

Функція активації (Activation Function) (рис. 3.3) - найбільш використовувана функція активації є RELU (Rectified Linear Unit), яка використовується як шар для видалення від'ємних значень з матриці значень, замінюючи їх на нуль. Вона зменшує лінійні можливості функції активації [25] без впливу на підобласті згорткових шарів. Через свою швидкодію при тренуванні моделі нейронних мереж для досягнення стандартної точності, функція RELU є бажаною. Найзагальніша формула виглядає наступним чином:

$$F(a) = \max(0, a) \quad (1)$$

де a є матрицею, що містить значення пікселів зображення. Використовувана сигмоїдна функція має наступний вигляд:

$$\sigma(a) = (1 + e^{-a})^{-1} \quad (2)$$

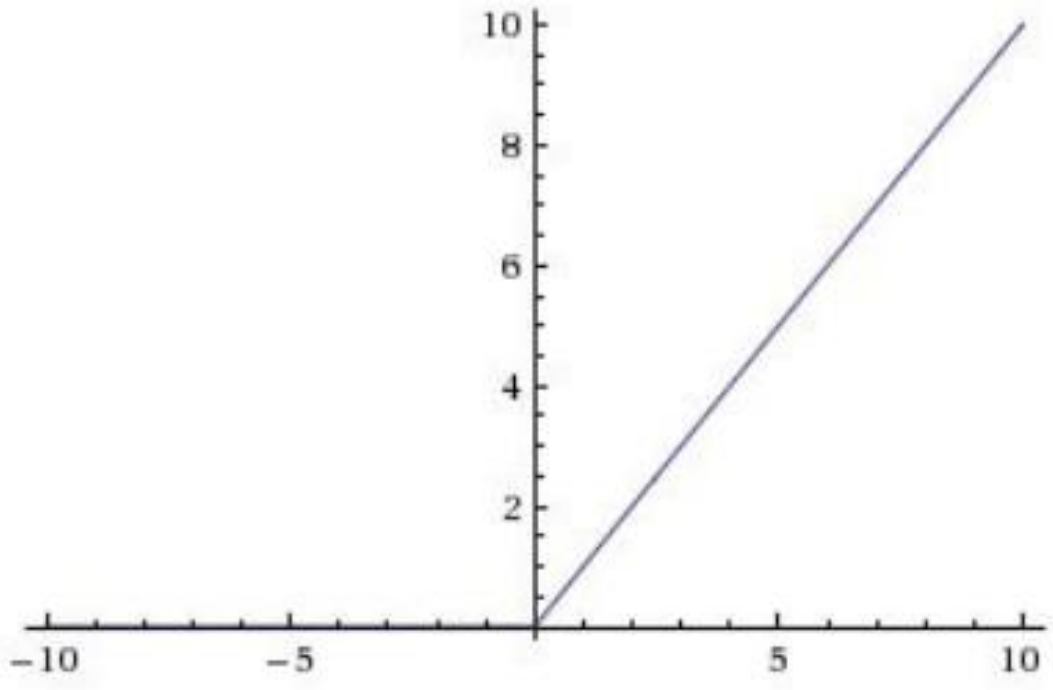


Рисунок 3.3 - Функція активації RELU

Dropout layer - Оскільки багато обмежень використовуються повністю зв'язаними шарами, можлива перенавчаність моделі. Один зі способів уникнути перенавчання - використання шару випадкового відкидання (dropout) [24]. Під час процесу навчання окремі елементи мережі видаляються з деякою ймовірністю (1- P) або залишаються з ймовірністю P. Ваги видалених елементів переналаштовуються для відповідних початкових ваг. Під час фаз навчання зазвичай використовується ймовірність 0.5 для відкидання зв'язків вхідних вузлів. Dropout допомагає зменшити перенавчання, якщо всі компоненти тренувальних даних були виключені. Цей підхід також значно покращує ефективність навчання.
Stochastic Pooling - Головним обмеженням Dropout є те, що вузли не повністю зв'язані з вузлами з попереднього сверточного шару. Для стохастичного пулінгу часто використовується стохастична техніка, яка замінює традиційні ймовірнісні пулінгові функції, де активація в кожній зоні пулінгу визначається випадковим чином на основі загального лінійного призначення, що залежить від активності в зоні пулінгу. (рис. 3.4)

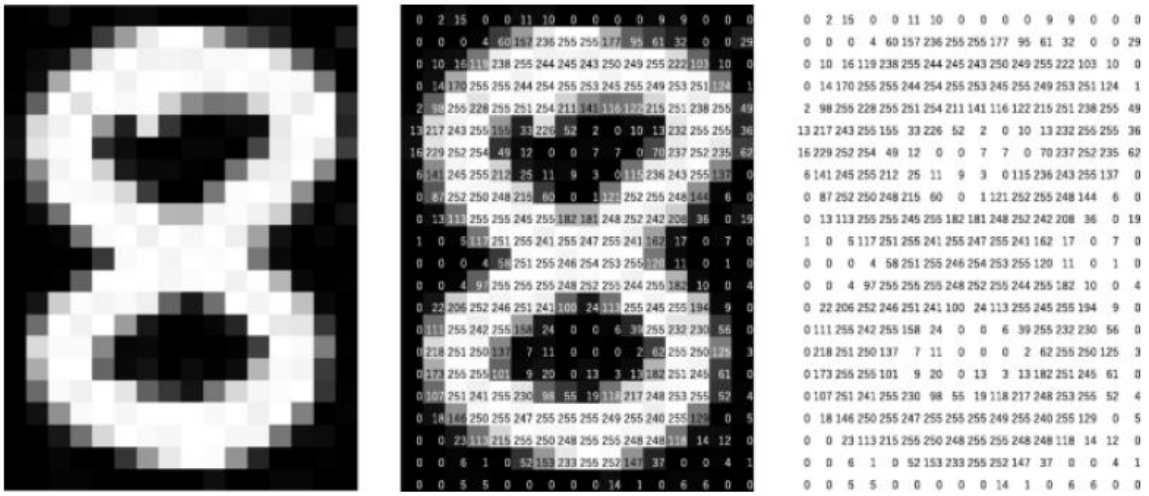


Рисунок 3.4 - Приклад надання чисельного значення кожному з пікселів після обробки шарів

У останніх дослідженнях, для виділення ознак використовувалися методи, які були створені вручну, наприклад, аналіз текстури, а потім використовувалися звичайні класифікатори машинного навчання, такі як випадкові ліси та машини

опорних векторів. [25] Однак, CNN має кілька відмінностей в порівнянні з цими методами. По-перше, CNN не потребує ручного виділення функцій. По-друге, архітектури CNN не вимагають обов'язкової сегментації пухлин або органів експертами. По-третє, CNN вимагає значно більше даних через велику кількість параметрів, які можуть бути навчені для оцінки. Це призводить до високих обчислювальних вимог і вимагає використання графічних процесорів (GPU) для навчання моделі.

Процес навчання вимагає багатьох ітерацій для правильного визначення потрібних частин. Жодна нейронна мережа не буде зі старту видавати правильний точний результат, але після процесу навчання можна отримати результати з точністю до 92% на зображеннях печінки (рис. 3.5)

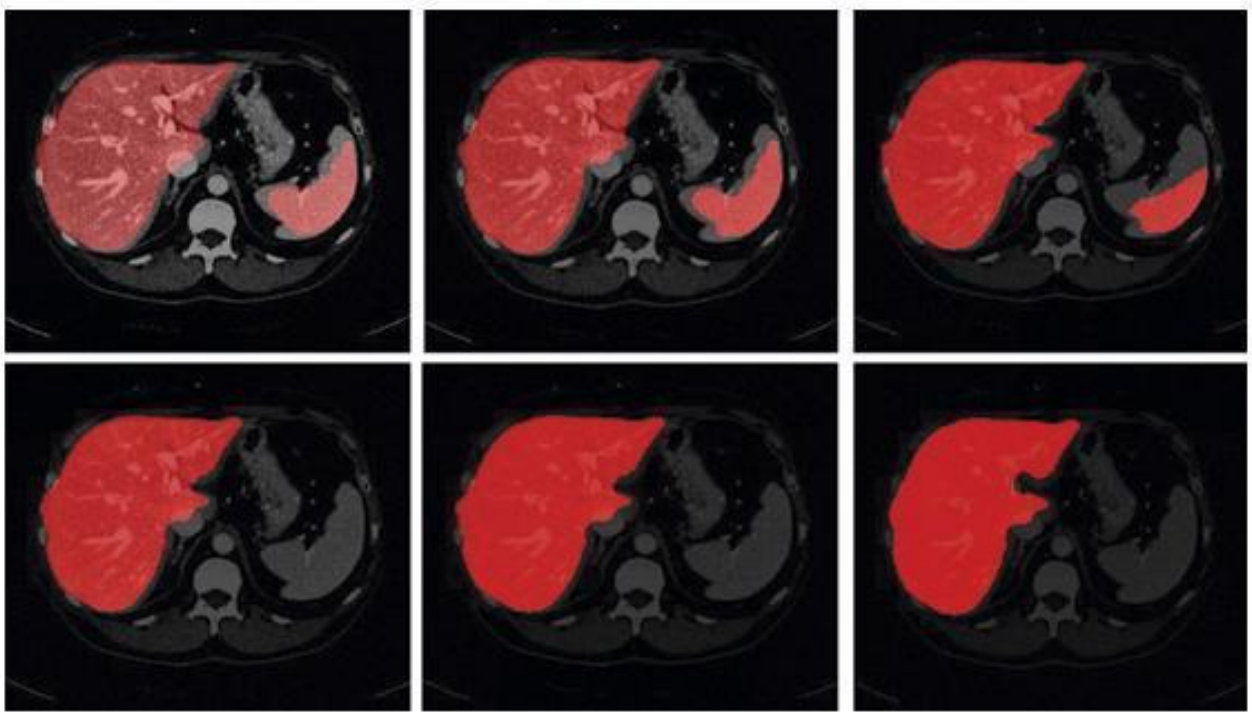
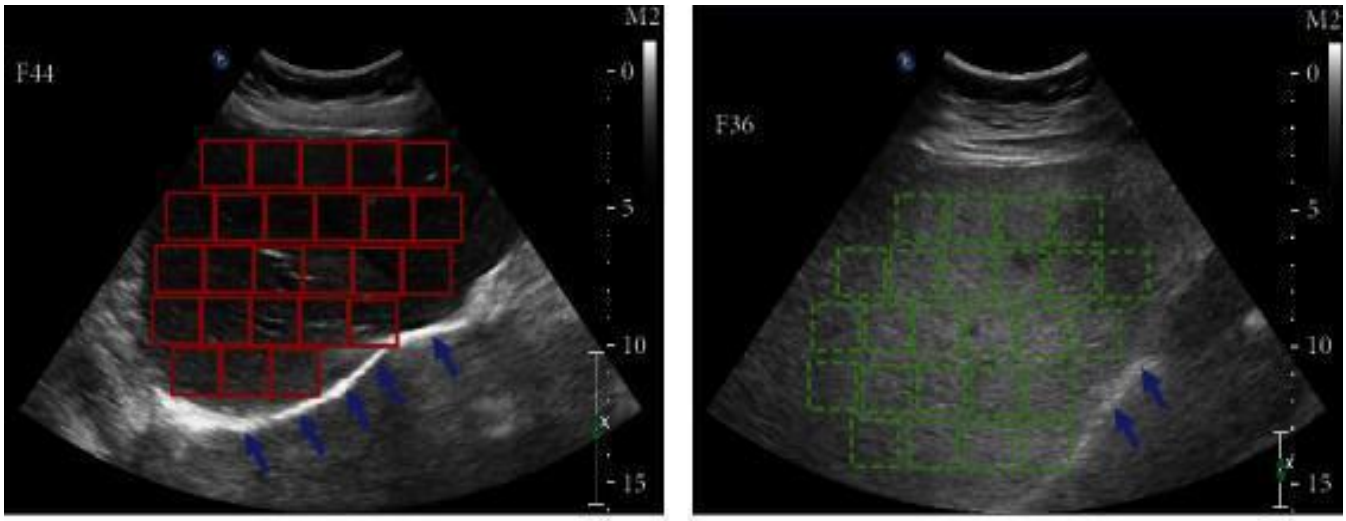


Рисунок 3.5 - Приклад навчання CNN моделі на томограмі печінки. Ітерації показані зверху зліва до низу справа: 8-ма, 15-та, 25-та, 55-та, 60-та та 70-та ітераційні карти ймовірності печінки. Світліша область вказує на більшу ймовірність належності до печінки.

Для навчання класифікатора зображень жирової дистрофії печінки ми використовуємо ультразвукове зображення нормальної печінки, жирової печінки низького ступеня, жирової печінки помірного ступеня і жирової печінки важкого ступеня, тобто 32 зображення з роздільною здатністю 1024 × 768. Близько 32 ультразвукові зображення отримані від 16 осіб, які складаються з 4 нормальних печінок, 4 жирових печінок низького ступеня, 4 жирових печінок помірного ступеня і 4 жирових печінок важкого ступеня. Для кожного ультразвукового зображення ми використовуємо одне рухоме вікно розміром w для розширення тренувальних зображень. На рис. 3.6 показано приклад класифікації зображень за допомогою анотацій ультразвукових зображень печінки. Кожен клас має приблизно 500 прикладів, загалом це становить близько 2000 тренувальних прикладів.



(a) (b)

Рисунок 3.6 - Приклад роботи нейромережі на УЗД печінки для визначення жирів

На рис. 3.7 - наведено кількісний і статистичний аналіз тренувальних прикладів чотирьох ультразвукових зображень печінки. C1, C2, C3 і C4 відповідають нормальній печінці, жировій печінці низького ступеня, жировій печінці помірного ступеня і жировій печінці важкого ступеня. Ми бачимо, що більшість патчів зображень мають різні інтервали сірого кольору. Червона лінія

відображає середнє значення сірого кольору, верхня чорна лінія - максимальне значення сірого кольору, а нижня чорна лінія - мінімальне значення сірого кольору.

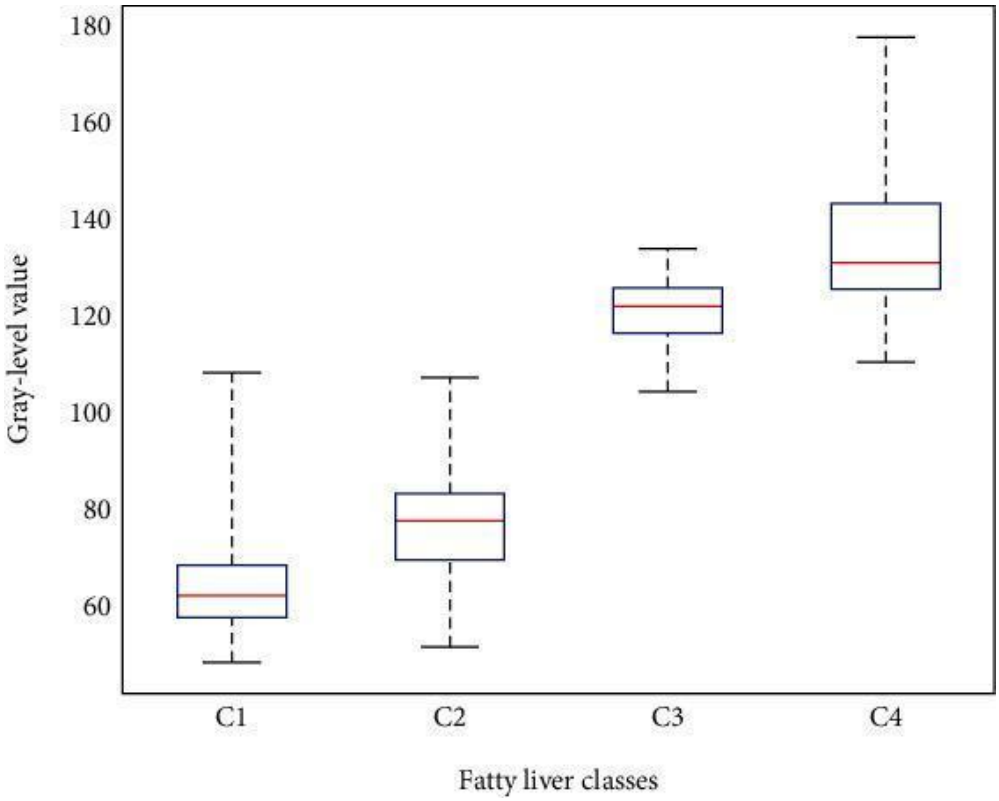


Рисунок 3.7 - Кількісний та статистичний аналіз тренувальних прикладів чотирьох ультразвукових зображень печінки.

3.1.2 Вейвлетні мережі

Вейвлетна нейронна мережа (Wavelet Neural Network, WNN) є комбінацією вейвлетного аналізу та штучних нейронних мереж. Вона поєднує вейвлети, які є математичними функціями, з нейронними мережами для обробки сигналів і аналізу даних.

Основна ідея вейвлетних нейронних мереж полягає в тому, що вейвлет-функції використовуються як базисні функції для розкладу сигналу чи зображення на рівні різної масштабності і частоти. Цей розклад дозволяє виявляти локальні особливості сигналу або зображення, такі як границі об'єктів чи деталі структури.

Працюючи з вейвлетними нейронними мережами, сигнал або зображення вводяться в мережу як вхідні дані. Ці дані проходять через шари нейронів, де вони піддаються обчисленням і операціям зваженого сумування. Кожен нейрон отримує вхідні сигнали, застосовує до них ваги та активаційну функцію і передає вихідний сигнал наступному шару нейронів.

Основна відмінність між вейвлетною нейронною мережею та згортковою нейронною мережею (Convolutional Neural Network, CNN) полягає в тому, як вони обробляють інформацію.

У вейвлетній нейронній мережі вейвлети використовуються як базисні функції для аналізу сигналів або зображень. Вейвлетні нейронні мережі можуть забезпечувати здатність адаптуватися до різних частот і просторових масштабів вхідних даних, оскільки вейвлети мають різні масштаби і зсуви. Вони показують високу ефективність при роботі з сигналами або зображеннями, що мають нелінійні структури і варіації масштабів.

Задано зображення x, ми можемо використовувати 2D-дискретне вейвлет-перетворення (2D-DWT) з чотирма фільтрами згортки: низькочастотним фільтром fLL і високочастотними фільтрами fLH, для розкладання x на чотири підзображення: xLL, xLH, xHL і xHH [26]. Зауважте, що чотири фільтри мають фіксовані параметри з згортковим кроком 2 під час перетворення. Взявши за приклад вейвлет Хаара, ці фільтри визначаються за формулою (1).

$$f_{LL} = \begin{bmatrix} +1 & +1 \\ +1 & +1 \end{bmatrix}, f_{LH} = \begin{bmatrix} -1 & -1 \\ +1 & +1 \end{bmatrix}, f_{HL} = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}, f_{HH} = \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \quad (1)$$

Крім того, операція 2D-дискретного вейвлет-перетворення (2D-DWT) визначається у формулі (2):

$$x_{LL} = (f_{LL} \otimes x) \downarrow_2, \quad x_{LH} = (f_{LH} \otimes x) \downarrow_2$$
$$x_{HL} = (f_{HL} \otimes x) \downarrow_2, \quad x_{HH} = (f_{HH} \otimes x) \downarrow_2 \qquad (2)$$

де $\otimes$ позначає оператор згортки, а $\downarrow 2$ означає стандартний оператор пониження розміру з коефіцієнтом 2. Іншими словами, математично DWT включає чотири фіксовані фільтри згортки з кроком 2 для реалізації оператора пониження розміру. Згідно з теорією перетворення Хаара [26], (i,j)-те значення xLL, xLH, xHL та xHH можна записати у формулі (3).

$$x_{LL}(i,j) = x(2i-1, 2j-1) + x(2i-1, 2j) + x(2i, 2j-1) + x(2i, 2j)$$
$$x_{LH}(i,j) = -x(2i-1, 2j-1) - x(2i-1, 2j) + x(2i, 2j-1) + x(2i, 2j)$$
$$x_{HL}(i,j) = -x(2i-1, 2j-1) + x(2i-1, 2j) - x(2i, 2j-1) + x(2i, 2j)$$
$$x_{HH}(i,j) = x(2i-1, 2j-1) - x(2i-1, 2j) - x(2i, 2j-1) + x(2i, 2j) \qquad (3)$$

Однією з переваг вейвлетних нейронних мереж є їх здатність розрізняти і відокремлювати деталі різних масштабів (рис. 3.8)

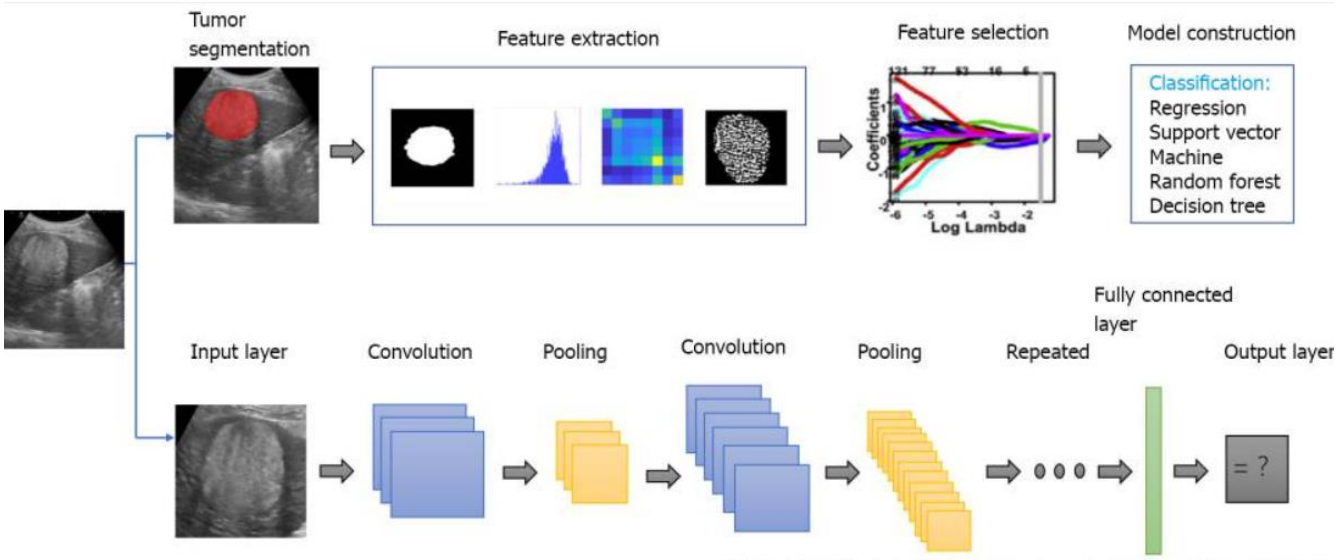


Рисунок 3.8 - Приклад розкладу картники двома різними згортками. Зверху – вейвлетна згортка, знизу – конволяційна згортка

3.2 Res Net глибока мережа

3.2.1 Чому обрано Res Net

Архітектура ResNet (Residual Network) є одним з найпопулярніших і успішних варіантів нейронних мереж для класифікації зображень. У цьому підрозділі обгрунтовується вибір архітектури ResNet для класифікації зображень УЗД печінки наступними аргументами:

- Глибокість мережі: ResNet вирішує проблему навчання глибоких нейронних мереж шляхом використання блоків зі з'єднаннями залишків. Це дозволяє створити мережу з десятками, а навіть сотнями шарів, зберігаючи при цьому ефективність та легкість навчання. Для класифікації складних зображень УЗД печінки, які можуть містити багато деталей та особливостей, важливо мати глибоку мережу, яка може виділяти й аналізувати ці характеристики.
- Зникаючий градієнт: Глибокі нейронні мережі зазвичай стикаються з проблемою зникаючого градієнта під час зворотного поширення помилки. ResNet вирішує цю проблему за допомогою з'єднань залишків, що дозволяють градієнтам легше прямувати вгору через мережу. Це полегшує процес навчання і допомагає досягти кращої збіжності.
- Пропускна здатність: ResNet має ефективну архітектуру, що дозволяє зменшити кількість параметрів мережі та обчислювальну складність, що знижує вимоги до ресурсів обчислювальної системи. Це особливо важливо для медичних досліджень, де може бути обмежена доступність обчислювальних ресурсів.
- Висока точність: ResNet показав вражаючі результати на багатьох важливих завданнях класифікації зображень. Використання ResNet для класифікації зображень УЗД печінки може допомогти досягти високої точності і надійності у визначенні різних станів та хвороб печінки.

Враховуючи всі ці аргументи, обрання архітектури ResNet є доцільним та обґрунтованим для класифікації зображень УЗД печінки у даній дипломній роботі.

3.2.2 Особливості архітектури Res Net

Одна з проблем, яку вирішують ResNet, - це відома проблема зникнення градієнту. Коли мережа стає занадто глибокою, градієнти, які обчислюються відносно функції втрати, легко зменшуються до нуля після кількох застосувань ланцюгового правила. Це призводить до того, що ваги ніколи не оновлюються, і, отже, навчання не відбувається. У ResNet градієнти можуть безпосередньо протікати через пропускні з'єднання в зворотному напрямку від пізніших шарів до початкових фільтрів.

У архітектурі ResNet (Residual Neural Network) використовується спеціальна концепція побудови блоків зі з'єднанням "скіп" (skip connections), що допомагає уникнути ефекту перенавчання. Суть полягає в передачі градієнтів безпосередньо через ці з'єднання.

Цей підхід до створення мережі зменшує ризик втрати градієнтів і сприяє більш стабільному тренуванню глибоких мереж. Замість того, щоб передавати градієнти тільки вперед, як це відбувається в звичайних мережах, градієнти можуть "пропливати назад" через шлях скіп-з'єднання, що дозволяє більш ефективну передачу інформації та зменшує ймовірність перенавчання.

Крім того, архітектура ResNet використовує пакетну нормалізацію (batch normalization) та шари з використанням активації ReLU, що сприяє стабільному навчанню та поліпшенню загальної продуктивності моделі.

Всі ці фактори разом сприяють зменшенню ефекту перенавчання та забезпечують кращу здатність узагальнення моделі на нові дані.

Головна ідея Res Net, яка відрізняє її від інших згорткових архітектур, це використання з'єднань залишків або "shortcut connections". Замість простого передачі вихідних значень шару наступному шару, ResNet додає "shortcut connections", які обходять кілька шарів. Ці з'єднання дозволяють пропускати шари та передавати вихідні значення безпосередньо на наступний шар, як показано на рис. 3.9 – рис. 3.10.

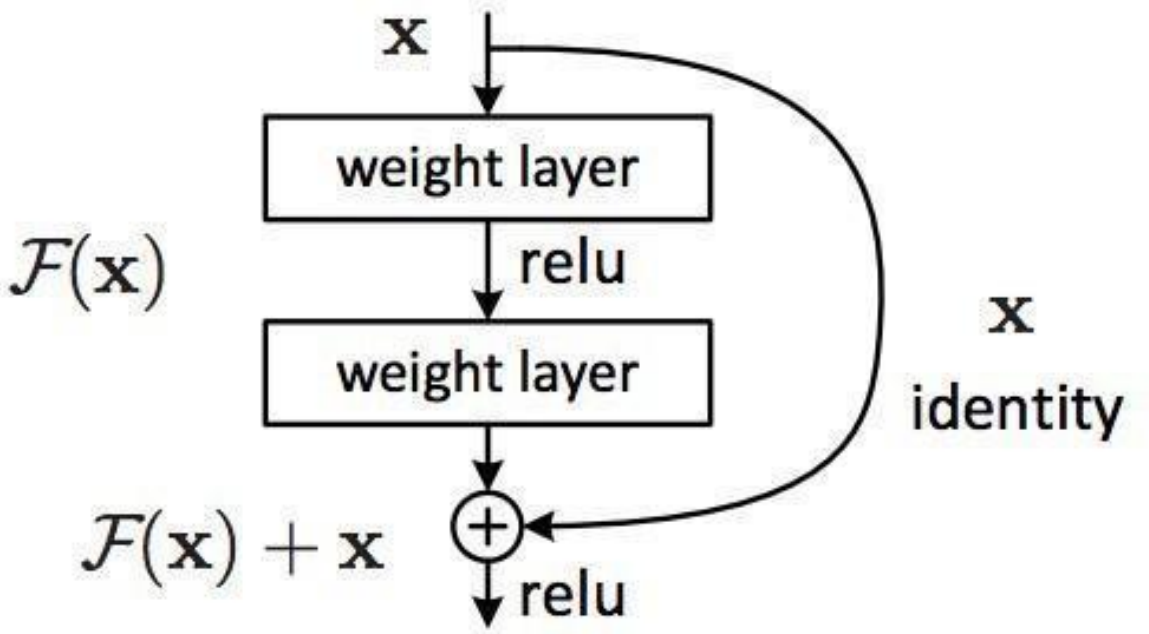


Рисунок 3.9 – схема передачі параметрів в архітектурі Res Net

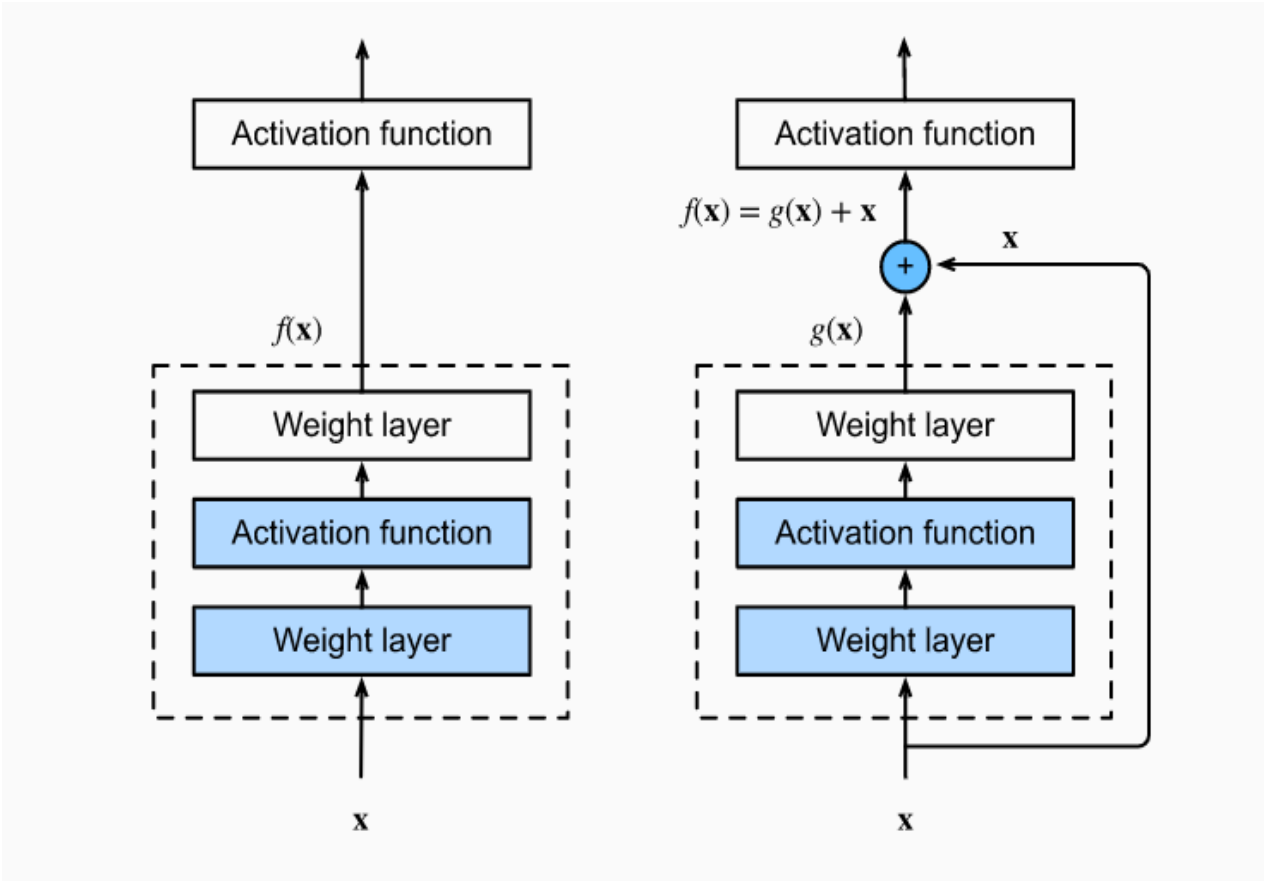


Рисунок 3.10. – схема передачі параметрів в архітектурі Res Net у порівнянні з класичною CNN

Стандартна архітектура ResNet має кілька варіантів, які відрізняються кількістю шарів і блоків. Один з найпопулярніших варіантів - ResNet-50, містить 50 шарів і включає конволюційні, пулінгові та повнозв'язані шари, а також вищезгадані резидуальні блоки.

У цій дипломній роботі для класифікації УЗД зображень печінки використовувалась наступна архітектура (рис 3.11):

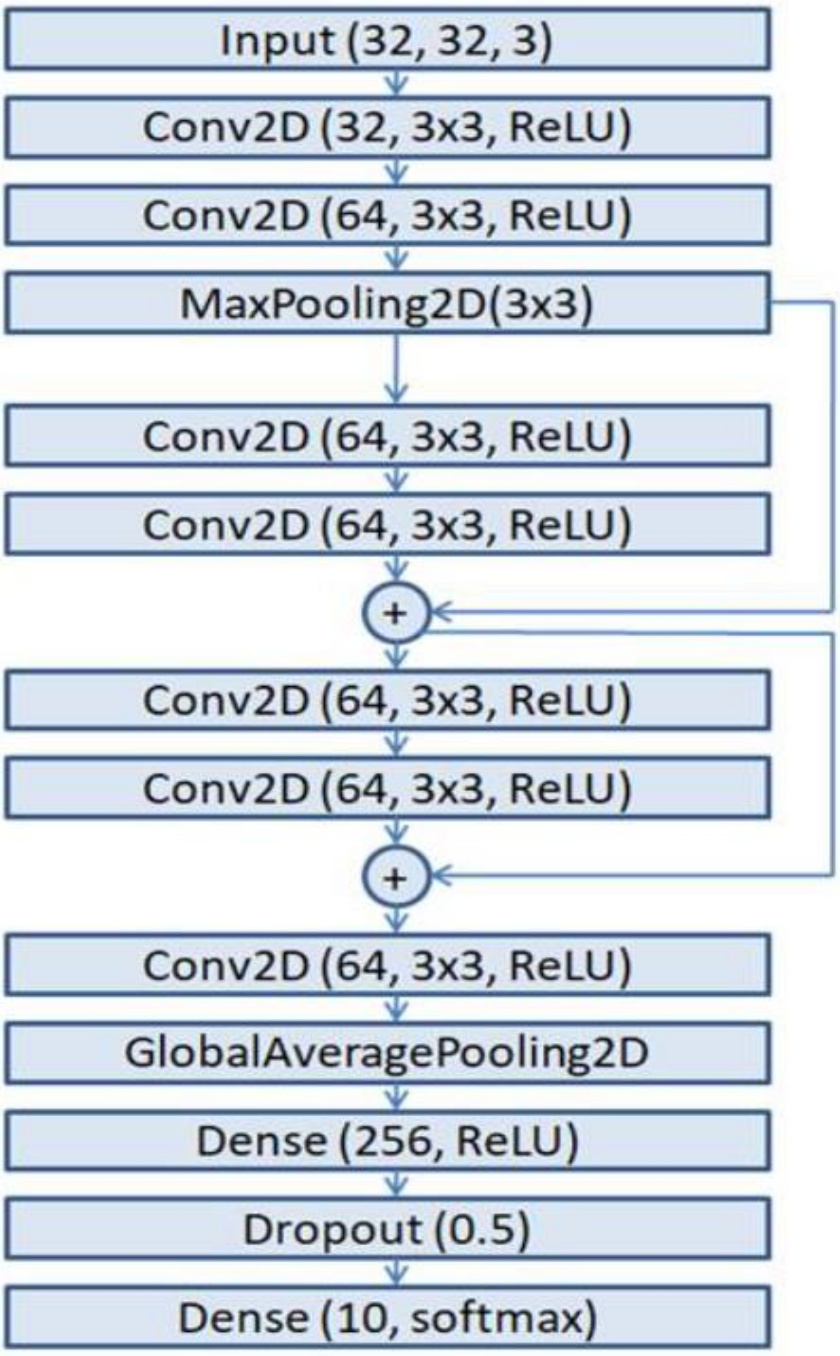


Рисунок 3.11 - Архітектура використаної мережі

У моделі є декілька етапів, що повторюються, використовуючи "резидуальні блоки". Загалом, кількість шарів в цій моделі залежить від кількості повторень цих блоків (рис. 3.12).

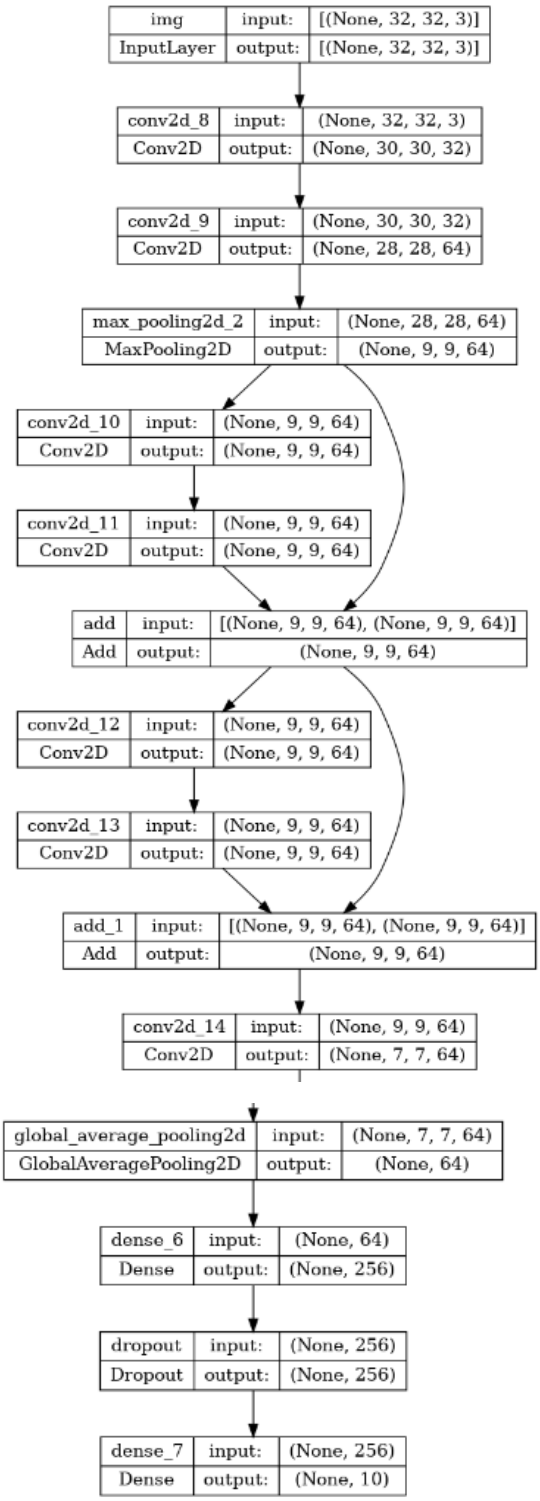


Рисунок 3.12 - Ретельний план роботи Res Net подібної мережі, що була застосована

У коді ці блоки мають таку послідовність:

Першим етапом є "вступний блок" (entry block), який має один сверточний шар (Conv2D) з 128 фільтрами, розміром ядра 3x3 і кроком 2. Після цього застосовується нормалізація пакета (BatchNormalization) і функція активації ReLU.

Потім повторюється наступна послідовність блоків: 3 блоки зі сверточними шарами (SeparableConv2D), застосовані послідовно, кожен з яких має фільтри розміром 3x3 і з BatchNormalization та ReLU після кожного з них. Після трьох таких блоків застосовується шар MaxPooling2D для зменшення розміру зображення.

Після цього повторюється аналогічна послідовність блоків, але з вищими розмірами фільтрів: 256, 512 і 728.

Наступний етап - окремий блок, що складається з окремого сверточного шару, BatchNormalization та ReLU.

Після цього використовується шар GlobalAveragePooling2D для згортання просторових розмірів зображення у середнє значення.

Залежно від кількості класів, модель використовує шар Dropout для запобігання перенавчання і повнозв'язаний шар (Dense) з відповідними функціями активації (sigmoid для двох класів або softmax для більш ніж двох класів).

Загалом, в цій моделі можна виокремити приблизно 48 сверточних шарів, що включають в себе окремі блоки і їхні повторення. Зважаючи на це, можна сказати, що вона має "ResNet-подібну" архітектуру з відносно великою кількістю шарів.

3.3 Підготовка датасету УЗД зображень

Початковий набір УЗД зображень складався з 21 зображення хворої печінки та 18 зображень здорової печінки розміром до 800 × 608 пікселів у форматі файлу JPG. Приклад на рис. 3.13.

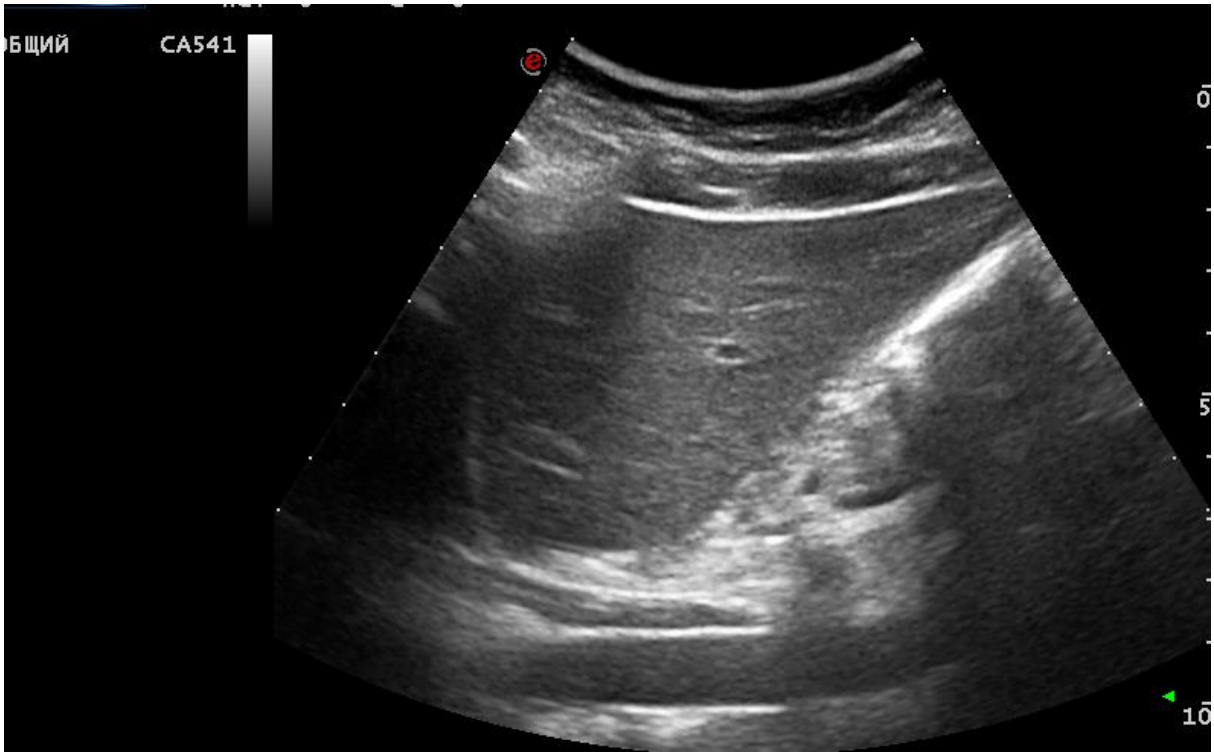


Рисунок 3.13 - Приклад зображення з тренувальної вибірки

Для збільшення розміру датасету та різноманітності даних було застосовано афінні перетворення до початкових зображень. Афінні перетворення включали зсуви, масштабування, обертання та відображення.

Афінні перетворення зображень - це група геометричних перетворень, які застосовуються до зображень для отримання нових варіацій або модифікації оригінального зображення. Ці перетворення змінюють положення, розмір, форму та орієнтацію зображення, забезпечуючи розширення датасету та збільшення його різноманітності.

Застосовані афінні перетворення включали:

- Зсув (Translation): Зсув зображення відбувається шляхом зміщення його пікселів у заданому напрямку та відстані. Наприклад, зсув вправо на 10 пікселів означає, що кожен піксель зображення буде зсунутий вправо на 10 пікселів.

- Масштабування (Scaling): Масштабування зображення відбувається шляхом зміни його розмірів. Зображення може бути зменшене (масштабування вниз) або збільшене (масштабування вгору) з врахуванням заданого коефіцієнта масштабування.
- Обертання (Rotation): Обертання зображення відбувається шляхом повороту його на певний кут навколо центру або заданої точки. Зображення може бути обернуте в годинниковому або проти годинникової стрілки напрямку.
- Відображення (Reflection): Відображення зображення відбувається шляхом віддзеркалення його вздовж горизонтальної або вертикальної осі. Це призводить до зміни положення та орієнтації об'єктів на зображенні.
- Зміщення (Shearing): Зміщення зображення відбувається шляхом розтягування або стискання зображення вздовж однієї з осей. Це може призводити до зсуву об'єктів на зображенні та зміни їх форми.

Загальний датасет, для тренування мережі, отриманий після розширення за допомогою афінних перетворень, складався з 502 зображень.

Потім з набору даних були видалені всі метадані та непотрібні чорні частини перед застосуванням гістограмної рівномірності (Histogram equalization) як показано на рис. 3.8. НЕ є корисною технікою обробки зображень, яка налаштовує інтенсивності зображення для покращення контрасту між медичними пристроями (рис. 3.14) [23]. Це дозволяє нашій глибокій нейронній моделі краще суміщуватись з зображеннями з різного обладнання або наявних баз даних.

Для оцінки ефективності нейронної мережі було виділено 5 зображень з початкового датасету для тестування. Більшу вибірку також було отримано шляхом застосування Афінних перетворень.

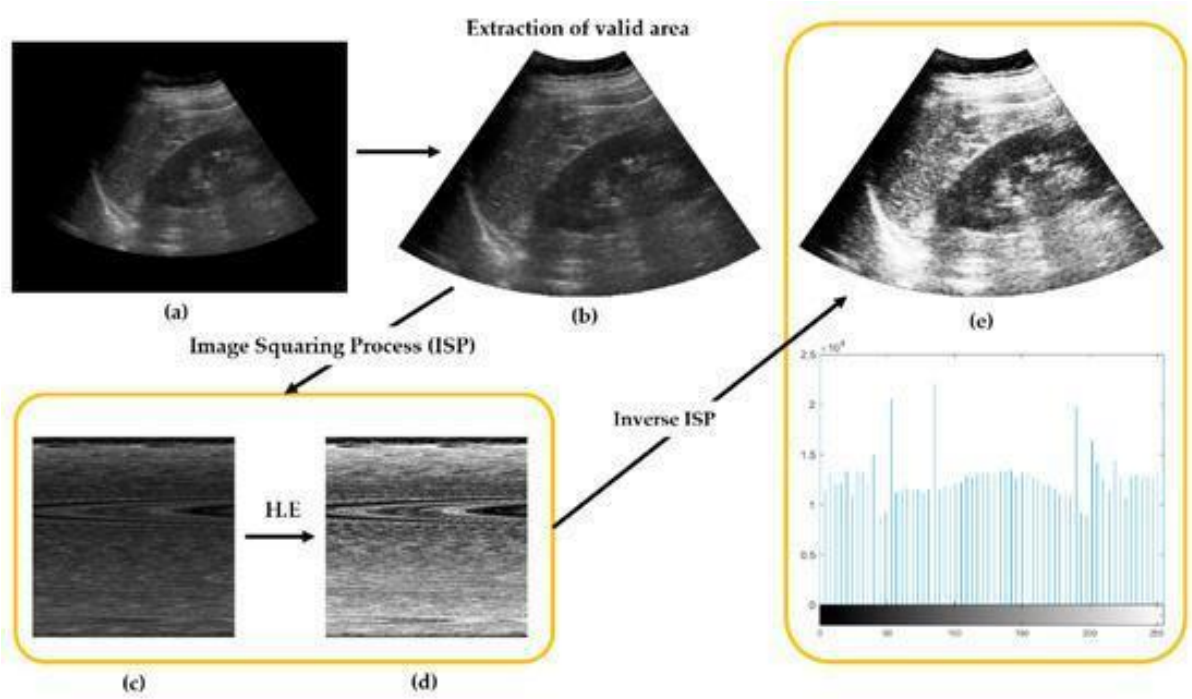


Рисунок 3.14 - а) Початкове зображення УЗД. б) Зображення, з якого видалені чорні частини. в) Зображення геометрично переструктуроване в квадратне зображення за допомогою ISP. Рівномірна розподіленість гістограми (HE) зображення з пункту (в). д) Зворотне перетворення зображення у початкову форму за допомогою зворотного ISP, та його гістограма. [34]

3.4 Вибір програмного середовища та інструментів

Для виконання даної роботи було обрано середовище програмування Python та досліджено наступні бібліотеки: TensorFlow, Keras, OpenCV, Pywavelets. Візуальним серидовищем програмування було обрано PyCharm.

3.4.1 Програмне середовище Python

У даному підрозділі визначимо чому було обрано саме мову програмування Python для реалізації поставлених задач.

Python сьогодні - це одна з найпопулярніших мов програмування для машинного навчання та розробки нейронних мереж. (рис. 3.15) **Низький поріг входу** дозволяє більшій кількості дата-дослідників швидко оволодіти Python і почати використовувати його для розробки штучного інтелекту без витрачання багато зусиль на вивчення мови. Python за структурою схожа на звичайну

англійську мову, що спрощує оволодіння нею. Її простий синтаксис дозволяє зручно працювати з складними системами, забезпечуючи зрозумілі зв'язки між елементами системи.

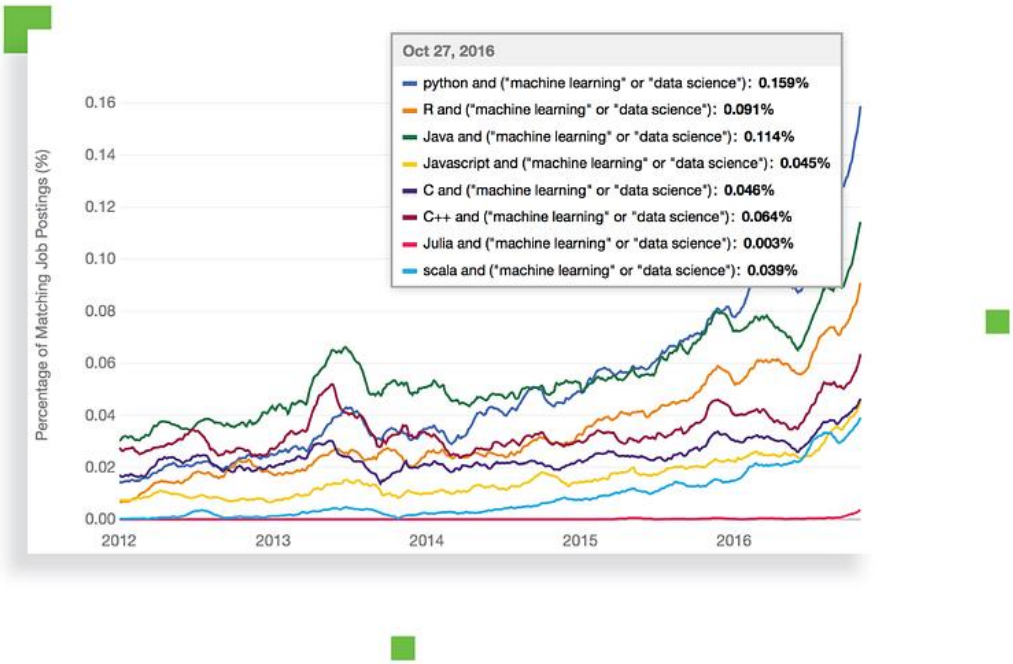


Рисунок 3.15 - Порівняння використання відомих мов програмування у машинному навчанні за Francois Puget [27]

Python має **широкий вибір практичних бібліотек**. Це одна з ключових причин його популярності для роботи зі штучним інтелектом Бібліотека - це модуль або група модулів, які опубліковані різними джерелами, такими як PyPi, і містять написаний наперед код, що дозволяє користувачам отримувати певні функціональні можливості або виконувати різні дії. Бібліотеки Python надають базові компоненти, щоб розробники не мусили починати кодування з самого початку кожного разу.

Машинне навчання вимагає постійної обробки даних, і бібліотеки Python дозволяють отримувати доступ до даних, їх обробку та трансформацію. Ось деякі популярні бібліотеки, які було використано в даній роботі:

- TensorFlow для роботи з глибоким навчанням, налаштування, навчання та використання штучних нейронних мереж з великими наборами даних.
- PyQt5: Бібліотека для створення графічного інтерфейсу користувача (GUI) в програмах на мові Python.
- Keras для глибокого навчання. Вона дозволяє швидкі обчислення і прототипування, оскільки використовує як процесор, так і графічний процесор (GPU) комп'ютера.
- OpenCV (cv2): Бібліотека комп'ютерного зору, яка надає функціональність для обробки зображень, зчитування та запису зображень.

Python не лише зручний у використанні та легкий у вивченні, але й дуже **універсальний**. Windows, Linux, MacOS, Unix та ще багато інших платформ підтримують запуск та розробку програм для розробки машинного навчання на цій мові. Для перенесення процесу з однієї платформи на іншу розробники повинні внести кілька невеликих змін і модифікувати деякі рядки коду, щоб створити виконуваний код для обраної платформи. Розробники можуть використовувати пакети, наприклад PyInstaller, для підготовки свого коду до запуску на різних платформах.

Внаслідок переваг, які були обговорені вище, Python стає все більш популярним серед дата-вчених. За даними StackOverflow, популярність Python передбачається зростати принаймні до 2023 року. (рис. 3.16) Це означає, що легше знайти розробників і замінити командних гравців за потреби. Крім того, вартість їх роботи може бути не такою високою, якщо використовувати менш популярну мову програмування.

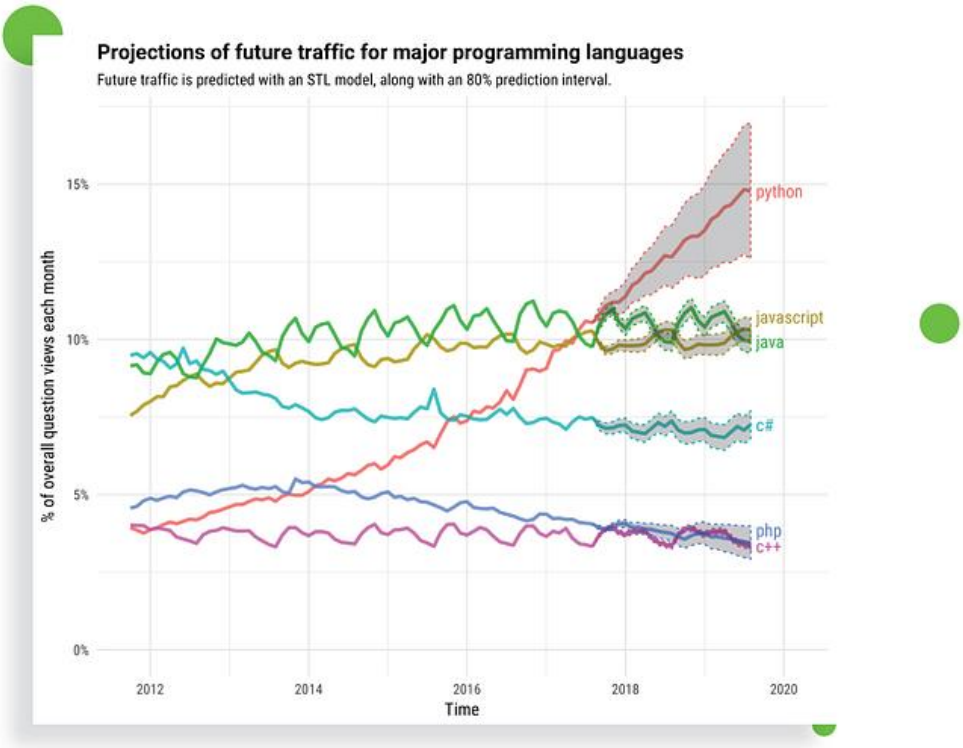


Рисунок 3.16 - Графік популярності мов програмування

3.4.2 Pywavelets

PyWavelets є потужним модулем для обробки медичних зображень, який надає різноманітні можливості. Основні можливості включають:

- Дискретне перетворення хвиль у різних просторових розмірах: PyWavelets дозволяє виконувати 1D та nD дискретне перетворення хвиль (DWT та IDWT), що дозволяє розкласти медичні зображення на різні компоненти за частотою.
- Стаціонарне перетворення хвиль: Модуль також підтримує стаціонарне перетворення хвиль, яке дозволяє зберегти деяні статистичні властивості сигналу після перетворення.
- Розклад та відновлення вейвлетного пакету: PyWavelets надає можливість розкладати та відновлювати сигнали за допомогою вейвлетного пакету, що дозволяє отримати додаткову інформацію про різні частотні компоненти зображення.

- Неперервне перетворення хвиль: Модуль також підтримує неперервне перетворення хвиль (CWT), що дозволяє аналізувати зображення в часово-частотному просторі.
- Вбудовані фільтри вейвлетів: PyWavelets містить більше 100 вбудованих фільтрів вейвлетів, а також підтримує створення власних фільтрів вейвлетів для специфічних дослідницьких або медичних потреб.
- Підтримка різних обчислювальних режимів: Модуль PyWavelets може виконувати обчислення з різною точністю, включаючи одинарну та подвійну точність, а також обчислення з використанням комплексних чисел.
- Сумісність з Matlab Wavelet Toolbox: Результати обробки за допомогою PyWavelets можуть бути сумісними з Matlab Wavelet Toolbox, що полегшує обмін результатами та порівняння алгоритмів обробки медичних зображень.

Вейвлетні перетворення також можуть використовуватися для обробки зображень тп передачі їх до нейронної мережі. (рис. 3.17 – рис. 3.18)

```
import numpy as np
import matplotlib.pyplot as plt

import pywt
import pywt.data


# Load image
original = pywt.data.camera()

# Wavelet transform of image, and plot approximation and details
titles = ['Approximation', ' Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = pywt.dwt2(original, 'bior1.3')
LL, (LH, HL, HH) = coeffs2
fig = plt.figure(figsize=(12, 3))
for i, a in enumerate([LL, LH, HL, HH]):
    ax = fig.add_subplot(1, 4, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=10)
    ax.set_xticks([])
    ax.set_yticks([])

fig.tight_layout()
plt.show()
```

Рисунок 3.17 - Приклад частини коду обробки зображення з використанням

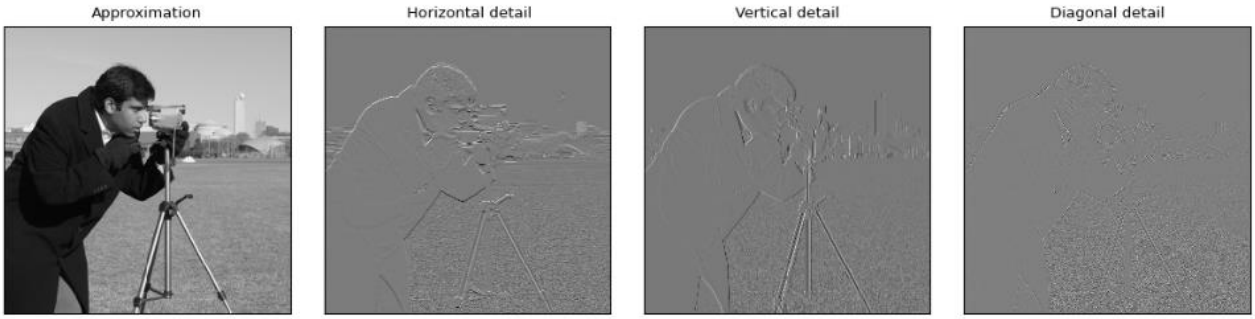


Рисунок 3.18 - Результат коду на рисунку 3.17

3.4.3 TensorFlow

Бібліотека TensorFlow є однією з найпопулярніших і потужних інструментів для розробки та реалізації нейронних мереж. Вона надає широкий спектр функціональних можливостей, що дозволяють зручно працювати з медичними зображеннями печінки та проводити їх аналіз.

У моїй роботі було вирішено використовувати TensorFlow для аналізу медичних зображень печінки з метою класифікації. Цей вибір є обґрунтованим з кількох причин:

- Гнучка архітектура: TensorFlow надає можливість реалізувати різні типи нейронних мереж, зокрема згорткові мережі, рекурентні мережі та глибокі нейронні мережі. Це дозволяє підібрати найбільш оптимальну архітектуру для класифікації медичних зображень печінки.

- Ефективна робота з великими наборами даних: TensorFlow забезпечує зручні інструменти для завантаження, обробки та попередньої обробки великих наборів медичних зображень. Це дозволяє ефективно працювати з великим обсягом даних та забезпечує точність і надійність результатів аналізу.
- Підтримка обчислення на графічних процесорах (GPU): TensorFlow може використовувати потужність графічних процесорів для прискорення обчислень, зокрема навчання глибоких нейронних мереж.

Це дозволяє швидше навчання моделей та більш швидке виконання класифікації зображень печінки.

- Широкий вибір функцій та модулів: TensorFlow має багатий набір функцій і модулів, що спрощує реалізацію різних алгоритмів та оптимізацію роботи з медичними зображеннями. Він також надає можливість використовувати готові моделі та переносити їх для власних потреб.

3.4.4 Keras

Keras — високорівневий API глибокого навчання, розроблений Google для впровадження нейронних мереж. Він написаний на Python і використовується для полегшення реалізації нейронних мереж. Він також підтримує обчислення кількох серверних нейронних мереж.

Keras забезпечує високорівневий інтерфейс для розробки нейронних мереж і є відносно легким у вивченні та використанні. Він надає Python API з високим рівнем абстракції та підтримує обчислення на кількох серверних елементах. Хоча це може зробити Keras трохи повільнішим порівняно з іншими фреймворками глибокого навчання, він є надзвичайно зручним для початківців. Крім того, Keras дозволяє переключатися між різними бекендами. Деякі з фреймворків, які підтримуються Keras, включають Tensorflow, Theano, PlaidML, MXNet і CNTK (Microsoft Cognitive Toolkit). (рис. 3.19)

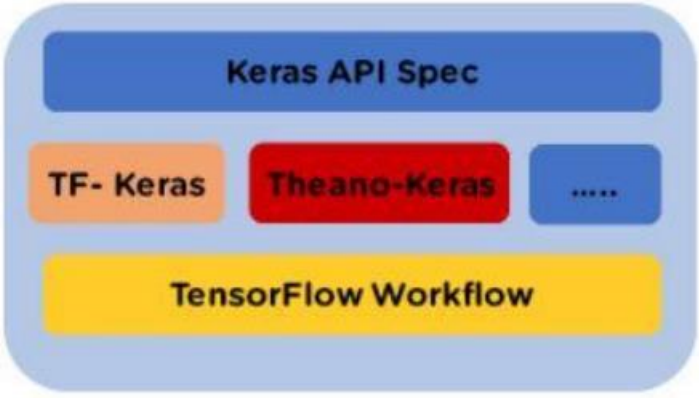


Рисунок 3.19 - Структура Keras

Keras є потужним інструментом, який допоміг успішно виконати дослідження та реалізувати нейронну мережу ResNet для класифікації УЗД зображень печінки у даній роботі. Використання Keras мало численні переваги, які значно полегшили розробку та навчання моделі. Деякі з них включають:

- Простота використання: У Keras досить зрозумілий та інтуїтивний інтерфейс, який легко використовувати. Завдяки цьому, можна швидко створити архітектуру нейронної мережі та налаштувати параметри шарів.
- Гнучкість: Keras надає різноманітні вбудовані шари та блоки, які дозволяють будувати різні архітектури нейронних мереж. Це дозволило легко використовувати блоки ResNet та налаштувати їх для вирішення поставленої конкретної задачі класифікації УЗД зображень печінки.
- Підтримка оптимізаторів та функцій втрат: Keras надає широкий вибір оптимізаторів та функцій втрат, що дозволяє вибрати оптимальні параметри для навчання моєї моделі.
- Зручне збереження та завантаження моделі: Keras дозволив зручно зберігати навчену модель після тренування у форматах, таких як HDF5, JSON або YAML. Це дозволило зберегти результати навчання для подальшого використання та класифікації нових зображень.

3.5 Реалізація програмного продукту

Почнемо будувати нейронну мережу для класифікації УЗД зображень печінки з імпортну головних бібліотек. (рис. 3.20)

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
```

Рисунок. 3.20 - Імпорт бібліотек

Ці рядки коду виконують наступне:

- import os: Імпортує модуль os для взаємодії з операційною системою.
- os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2': Встановлює змінну середовища, щоб приховати повідомлення про рівень журналу TensorFlow.
- import tensorflow as tf: Імпортує бібліотеку TensorFlow для створення і навчання моделі.
- from tensorflow import keras: Імпортує підмодуль keras з TensorFlow для побудови моделі нейронної мережі.
- from tensorflow.keras import layers: Імпортує класи шарів для будівництва моделі.
- import matplotlib.pyplot as plt: Імпортує бібліотеку matplotlib для візуалізації зображень.

Далі візьмемо попередньо оброблені ультразвукові зображення щоб завантажити їх для навчання і тренування нейромережі (рис. 3.21)

```
plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
    for i in range(6):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(int(labels[i]))
        plt.axis("off")
        plt.show()
```

Рисунок 3.21 - Візуалізація даних

Цей код створює графічний вивід для перегляду зображень з тренувального набору даних. Зображення та відповідні мітки відображаються у форматі 3x3 на графіку.

Розглянемо докладніше функцію make_model, яка будує модель нейронної мережі з використанням архітектури ResNet. Функція приймає форму вхідних даних input_shape і кількість класів num_classes, і повертає об'єкт моделі.

Далі створюється об'єкт моделі model з використанням функції make_model, передаючи розмір вхідного зображення та кількість класів. (рис. 3.22)

```
def make_model(input_shape, num_classes):
    inputs = keras.Input(shape=input_shape)
        # Entry block
    x = layers.Rescaling(1.0 / 255)(inputs)
```

Рисунок 3.22 – Оголошення функції створення нейромережі

У перших рядках функція make_model оголошує вхідний шар моделі за допомогою keras.Input. input_shape вказує форму вхідних зображень, а inputs є вхідним тензором. Потім використовується шар layers.Rescaling, щоб відмасштабувати значення пікселів зображення. Зображення поділяються на 255, щоб нормалізувати їх значення до діапазону [0, 1]. Результат присвоюється змінній x.

Далі виконується первинна згортка (рис. 3.23):

```
    x = layers.Conv2D(128, 3, strides=2, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    previous_block_activation = x  # Set aside residual
```

Рисунок 3.23 – Перші шари мережі

У цих рядках використовуються шари layers.Conv2D, layers.BatchNormalization та layers.Activation для виконання свертки, нормалізації та активації послідовно. Застосовується свертка з 128 фільтрами розміром 3x3, з кроком 2 та заповненням "same". Після свертки виконується нормалізація та активація за допомогою функцій BatchNormalization та "relu" відповідно.

Додаємо власне шари притаманні архітектурі Res Net (рис. 3.24)

```
for size in [256, 512, 728]:
    x = layers.Activation("relu")(x)
    x = layers.SeparableConv2D(size, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)

    x = layers.Activation("relu")(x)
    x = layers.SeparableConv2D(size, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)

    x = layers.MaxPooling2D(3, strides=2, padding="same")(x)

    # Project residual
    residual = layers.Conv2D(size, 1, strides=2, padding="same")(
        previous_block_activation
    )
    x = layers.add([x, residual])  # Add back residual
    previous_block_activation = x  # Set aside next residual

x = layers.SeparableConv2D(1024, 3, padding="same")(x)
x = layers.BatchNormalization()(x)
```

Рисунок 3.24 – Код архітектури моделі

У цьому фрагменті коду виконується створення блоків з використанням сверток. Цикл перебирає розміри [256, 512, 728] та для кожного розміру застосовує наступні шари:

- Активація "relu"
- Роздільна свертка (SeparableConv2D) з кількістю фільтрів size, розміром 3x3 та заповненням "same"
- Нормалізація за допомогою BatchNormalization
- Знову активація "relu"
- Ще одна роздільна свертка з тими ж параметрами
- Ще одна нормалізація
- Далі виконується шар layers.MaxPooling2D з розміром пулінгу 3x3, кроком 2 та заповненням "same".

Також у циклі використовується поняття "residual". Значення x додається до значення попереднього блоку residual за допомогою шару layers.add. Це допомагає

передати інформацію безпосередньо від одного блоку до наступного, дозволяючи нейромережі навчатися ефективніше.

Після циклу блоків застосовується ще один шар layers.SeparableConv2D з 1024 фільтрами, розміром 3x3 та заповненням "same". Його вихід нормалізується та активується за допомогою BatchNormalization та активації "relu".

Після останнього шару SeparableConv2D використовується шар layers.GlobalAveragePooling2D для згортання просторових розмірів тензора в середні значення.

Також визначаються параметри для остаточного шару, який залежить від кількості класів num_classes. Якщо num_classes дорівнює 2, то використовується активація "sigmoid" та один вихідний нейрон, інакше використовується активація "softmax" та num_classes вихідних нейронів.

Додатково застосовується шар layers.Dropout для регуляризації шляхом випадкового вимкнення 50% нейронів.

Нарешті, визначається вихід моделі і повертається створена модель keras.Model з вхідними inputs та вихідними outputs. (рис. 3.25)

```
x = layers.SeparableConv2D(1024, 3, padding="same")(x)
x = layers.BatchNormalization()(x)
x = layers.Activation("relu")(x)

x = layers.GlobalAveragePooling2D()(x)
if num_classes == 2:
    activation = "sigmoid"
    units = 1
else:
    activation = "softmax"
    units = num_classes

x = layers.Dropout(0.5)(x)
outputs = layers.Dense(units, activation=activation)(x)
return keras.Model(inputs, outputs)
```

Рисунок 3.25 – Вихідні шари мережі

layers.Conv2D: Шар згортки виконує згортку зображення з фільтрами згортки. Формула для шару згортки на один вихідний канал має вигляд (3.26):

$$Y(i,j,k) = \sum_{s,t,r,s,t} W(r,s,k) \cdot X(i+s, j+t, r)$$

Рисунок 3.26 - Формула шару згортки

Де:

Y(i, j, k) - вихідне значення згортки в точці (i, j) для k-го вихідного каналу,

W(r, s, k) - вага згортки для r-го рядка, s-го стовпця та k-го вихідного каналу,

X(i + s, j + t, r) - вхідне значення в точці (i + s, j + t) для r-го вхідного каналу.

layers.BatchNormalization: Шар нормалізації партій нормалізує вихідні значення шару передачі далі. Формула для шару нормалізації партій має вигляд (рис.3.27):

$$\hat{X}(i,j,k) = \frac{X(i,j,k) - \mu_k}{\sqrt{\sigma_k^2 + \epsilon}} \cdot \gamma_k + \beta_k$$

Рисунок. 3.27 - Формула для шару нормалізації

Де:

X(i, j, k) - вхідне значення в точці (i, j, k),

$\hat{X}(i,j,k)$ - нормалізоване вихідне значення в точці (i, j, k),

μ_k - середнє значення для k-го каналу,

σ_k - стандартне відхилення для k-го каналу,

ϵ - невелика константа для числової стійкості,

γ_k - масштабуючий коефіцієнт для k-го каналу,

- - зсув для k-го каналу.

3.6 Аналіз та порівняння одержаних результатів

Після запуску коду нейронної мережі, в терміналі можемо побачити наступні записи, які описують процес тренування нашої моделі. (рис. 3.28)

```
Found 5016 files belonging to 2 classes.
Using 4013 files for training.
Using 1003 files for validation.
502/502 [==============================] - 148s
274ms/step - loss: 0.1031 - accuracy: 0.9604 - val_loss: 2.2235 - val_accuracy: 0.6251
Epoch 2/4
502/502 [==============================] - 135s
268ms/step - loss: 0.0287 - accuracy: 0.9890 - val_loss: 0.5017 - val_accuracy: 0.8076
Epoch 3/4
502/502 [==============================] - 141s
281ms/step - loss: 0.0266 - accuracy: 0.9913 - val_loss: 3.8058 - val_accuracy: 0.6251
Epoch 4/4
502/502 [==============================] - 139s
276ms/step - loss: 0.0065 - accuracy: 0.9993 - val_loss: 2.4277e-04 - val_accuracy: 1.0000
```

Рисунок 3.28 – звіт протягом тренування моделі

Метрики розшифровуються таким чином:

- Loss (втрати): У першу епоху втрати (loss) становлять 0.1031 на тренувальному наборі та 2.2235 на наборі валідації. Чим нижче значення втрати, тим краще модель навчається передбачати правильні мітки. На першій епосі втрати впали з 2.2235 до 0.1031 на наборі валідації, що свідчить про покращення моделі.
- Accuracy (точність): У першій епосі точність складає 0.9604 на тренувальному наборі та 0.6251 на наборі валідації. Точність вимірює, яку частку зображень модель класифікувала правильно. Значення точності близько до 1.0 вказує на хорошу натренованість моделі. Тут ми отримали високу точність на тренувальному наборі, але значення на наборі валідації нижче, що може свідчити про перенавчання моделі на тренувальних даних.

- Val_loss (втрати на валідації): На четвертій епосі втрати на наборі валідації складають 2.4277e-04. Дуже низьке значення втрати означає, що модель дуже точно передбачає класи зображень на наборі валідації.
- Val_accuracy (точність на валідації): На четвертій епосі точність на наборі валідації досягає 1.0, що означає, що модель дуже точно класифікує зображення на валідаційному наборі.

На потужному ПК одна епоха тренування займала в середньому трохи більше двох хвилин. Варто зазначити що на слабшому пк з 8 гб оперативної пам’яті без затосування графічного процессора тренування такої моделі займає набагато більше часу (кілька годин), а на ще слабшому може взагалі зупиняти роботу системи!

Загалом, результати показують, що модель ефективно навчається розпізнавати зображення УЗД печінки. Зменшення втрат та зростання точності на тренувальному та валідаційному наборах свідчить про покращення моделі з кожною епохою. Досягнення точності 1.0 на валідаційному наборі вже на четвертій епосі означає, що модель добре узагальнює та добре працюватиме на нових зображеннях печінки.

Тепер ми можемо звертатися до натренованої моделі в окремому файлі. Запускаємо розпізнавання зображень із патологією. Нейромережа розпізнає зображення ураженої печінки з високою точністю. Результат продемонстровано на рис. 3.29.

Рисунок 3.29 – Результат класифікації зображень з патологією

Так само подамо на вхід мережі кілька зображень здорової печінки(рис. 3.30):

```
$ python pred.py /home/wit/PycharmProjects/liver/Images/ManAug/Test/0
['_20181030102522_1029060.jpg', '_20181030102522_1026420.jpg', '_20181030102522_1029230.jpg']
_20181030102522_1029060.jpg
1/1 [==============================] - 2s 2s/step
Image _20181030102522_1029060.jpg:
 is 100.00% Good and 0.00% Bad.
_20181030102522_1026420.jpg
1/1 [==============================] - 0s 17ms/step
Image _20181030102522_1026420.jpg:
 is 45.48% Good and 54.52% Bad.
_20181030102522_1029230.jpg
1/1 [==============================] - 0s 18ms/step
Image _20181030102522_1029230.jpg:
 is 100.00% Good and 0.00% Bad.
```

Рисунок 3.30 – Результат класифікації зображень здорової печінки

Цікаво що у тестових даних завалялось зображення ураженої печінки в папці, де лежать зображення здорової. На рисунку 3.30, Image 2 нейромережа розпізнає зображення з упевненістю 54.52% як уражене патологією. (Рис. 3.31)

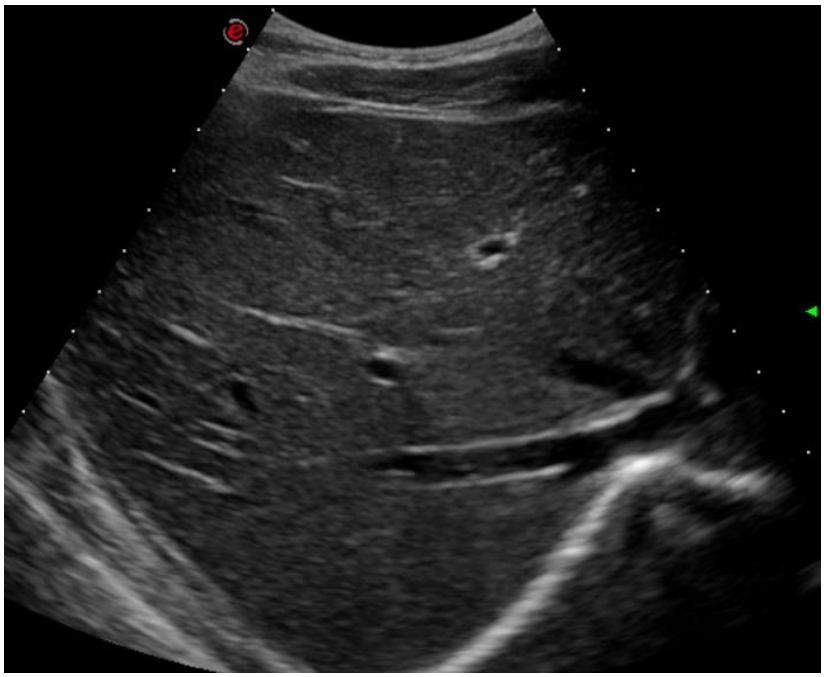


Рисунок 3.31 – Зображення печінки з тестових даних

Протестуємо також нейромережу на іншому ПК. Тепер, коли тренування вже було виконано, можна переносити модель у вигляді папки з файлами і вказувати її

розміщення. Ось ще приклади класифікації зображень УЗД печінки на тестових зображеннях (рис. 3.32 – рис. 3.33):

```
D:\Univer\CNN\TestImages\ShiftedH\h-2___20181030102522_1026420.jpg
1/1 [==============================] - 3s 3s/step
Image h-2___20181030102522_1026420.jpg:
 is 51.95% Good and 48.05% Bad.
h-2___20181030102522_1029060.jpg
D:\Univer\CNN\TestImages\ShiftedH\h-2___20181030102522_1029060.jpg
1/1 [==============================] - 0s 265ms/step
Image h-2___20181030102522_1029060.jpg:
 is 100.00% Good and 0.00% Bad.
h-2___20181030102522_1029230.jpg
D:\Univer\CNN\TestImages\ShiftedH\h-2___20181030102522_1029230.jpg
1/1 [==============================] - 0s 249ms/step
Image h-2___20181030102522_1029230.jpg:
 is 100.00% Good and 0.00% Bad.
h-2_____________20181030091812_0922420.jpg
D:\Univer\CNN\TestImages\ShiftedH\h-2_____________20181030091812_0922420.jpg
1/1 [==============================] - 0s 242ms/step
Image h-2_____________20181030091812_0922420.jpg:
 is 0.10% Good and 99.90% Bad.
h-2_____________20181030091812_0922510.jpg
D:\Univer\CNN\TestImages\ShiftedH\h-2_____________20181030091812_0922510.jpg
```

Рисунок 3.32 – Результати роботи мережі на слабшому ПК

```
Image h-4___20181030102522_1029230.jpg:
 is 100.00% Good and 0.00% Bad.
h-4_____________20181030091812_0922420.jpg
D:\Univer\CNN\TestImages\ShiftedH\h-4_____________20181030091812_0922420.jpg
1/1 [==============================] - 0s 245ms/step
Image h-4_____________20181030091812_0922420.jpg:
 is 0.15% Good and 99.85% Bad.
h-4_____________20181030091812_0922510.jpg
D:\Univer\CNN\TestImages\ShiftedH\h-4_____________20181030091812_0922510.jpg
1/1 [==============================] - 0s 247ms/step
Image h-4_____________20181030091812_0922510.jpg:
 is 0.00% Good and 100.00% Bad.
h2___20181030102522_1026420.jpg
D:\Univer\CNN\TestImages\ShiftedH\h2___20181030102522_1026420.jpg
1/1 [==============================] - 0s 261ms/step
Image h2___20181030102522_1026420.jpg:
 is 35.95% Good and 64.05% Bad.
h2___20181030102522_1029060.jpg
D:\Univer\CNN\TestImages\ShiftedH\h2___20181030102522_1029060.jpg
1/1 [==============================] - 0s 250ms/step
Image h2___20181030102522_1029060.jpg:
 is 100.00% Good and 0.00% Bad.
h2___20181030102522_1029230.jpg
D:\Univer\CNN\TestImages\ShiftedH\h2___20181030102522_1029230.jpg
1/1 [==============================] - 0s 237ms/step
Image h2___20181030102522_1029230.jpg:
 is 100.00% Good and 0.00% Bad.
h2_____________20181030091812_0922420.jpg
```

Рисунок 3.33 – Результати роботи мережі на слабшому ПК

Можемо бачити 100% або близьку до цього точність визначення на більшості зображеннях. Як правило ті зображення, де точність невизначена – нечіткі самі по собі, але модель всеодно схиляється до правильної відповіді.(рис. 3.34)

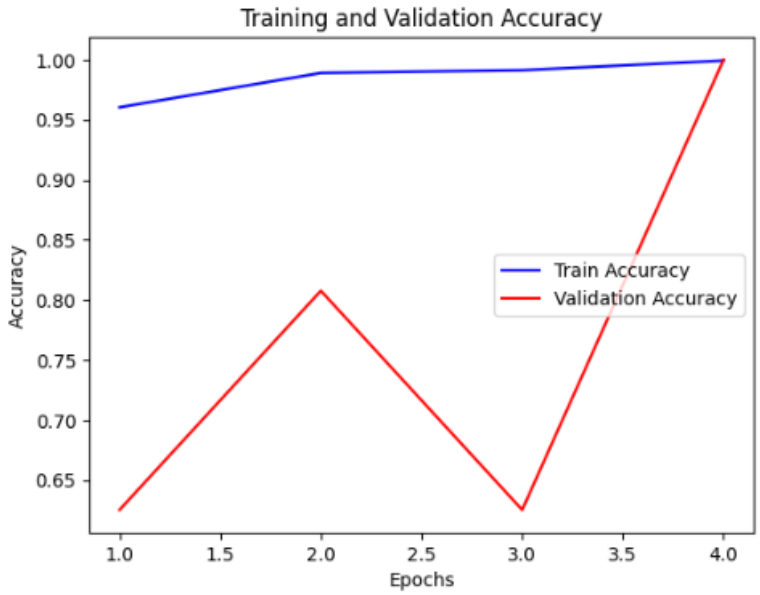


Рисунок 3.34 – Графік зміни точності тренувального та валідаційного наборів з плином епох

3.7. Практичне застосування

На практиці для класифікації медичних зображень застосовуються різноманітні варіації архітектури ResNet, зокрема ResNet-50, ResNet-101, та ResNet-152 (табл. 3.1.).

Основні області застосування включають:

- Лікарні та клініки: ResNet та інші нейронні мережі використовуються для автоматичної класифікації та діагностики медичних зображень, таких як рентгенівські знімки, комп'ютерна томографія (CT), магнітно-резонансна томографія (MRI) тощо. Вони допомагають лікарям виявляти ознаки захворювань, ракові пухлини, аномалії та допомагають в прийнятті рішень щодо лікування.

- Медичні дослідницькі лабораторії: Нейронні мережі використовуються для обробки та аналізу великих обсягів медичних зображень у дослідженнях. Вони допомагають у виявленні нових ознак, зв'язків та шаблонів, що можуть сприяти розвитку нових методів діагностики та лікування.
- Системи телемедицини: У віддалених районах або під час телеконсультацій нейронні мережі використовуються для класифікації та аналізу медичних зображень, які передаються в реальному часі.

Таблиця 3.1

Порівняння різних нейромереж для класифікації зображень

Тип нейромережі	**Точність (%)**	**Чутливість (%)**	**Специфічність (%)**	**Час навчання (год)**
Згорткові мережі	92	89	93	12
Вейвлетні мережі	87	85	88	10
Рекурентні нейромережі	90	88	91	15
Конволюційні рекурентні нейромережі	94	91	95	18
Трансформери	93	90	94	20
Глибокі нейромережі з передобробкою даних	95	92	96	14

3.8. Розрахунок економічного ефекту

Після проведення економічного аналізу та врахування теоретичних знань з економіки та організації виробництва, було встановлено чотири потенційні варіанти реалізації програмного продукту на основі його основних функцій. За

результатами оцінки техніко-економічного рівня, найбільш ефективним виявився перший варіант, який забезпечує найвищий коефіцієнт техніко-економічного рівня.

Цей обранний варіант включає в себе використання мови програмування Python, використання зображень печінки для тренування нейромережі, використання бібліотеки pywavelets для покращення точності розрахунків, інтерфейс користувача, побудований на базі Tensor Flow, а також досягнення найвищої точності розрахунків.

Загальна вартість витрат для цього варіанту становить 220 111,42 грн. Обрання цього варіанту передбачає оптимальне використання ресурсів і потенціалу програмного продукту, що дозволить досягти максимального економічного ефекту.

Висновки до розділу 3

У цьому практичному розділі ми провели аналіз зображень УЗД печінки з використанням різних нейронних мереж. Було обрано датасет зображень та створено більше вибірку з допомогою афінних перетворень. Було проведено класифікацію зображень печінки з допомогою нейромережі з поділом на здорову (Liver Good) та уражену патологією (Liver Bad).

У результаті ми отримали натренеровану згорткову мережу, яка класифікує узд зображення печінки з майже 100% точністю. Такий результат було отримано з використанням 4013 зображень для тренування та 1003 для тестування мережі. Було проведено 3 епохи для отримання такої точності.

ЗАГАЛЬНІ ВИСНОВКИ

У даній дипломній роботі було поставлено на меті розробку та навчання штучної нейронної мережі для класифікації медичних зображень УЗД печінки з метою покращення діагностики хвороб печінки. Для досягнення цієї мети були визначені наступні завдання:

1. Проведено аналіз вітчизняних та зарубіжних джерел, що дозволило ознайомитися з основними теоретичними аспектами класифікації медичних зображень та різними підходами до використання нейромереж в цій області.
2. Зібрано та підготовлено базу даних медичних зображень УЗД печінки для подальшого навчання та тестування моделей. Цей крок включав у себе обробку зображень, вирівнювання розмірів, розбиття на навчальний та тестовий набори даних.
3. Розроблено модель штучної нейронної мережі ResNet для виконання класифікації УЗД зображень печінки. Була використана бібліотека Keras для реалізації моделі та навчання з використанням набору даних.
4. Проведено експерименти та тестування розробленої моделі на тестових даних. Отримані результати показали високу точність та ефективність моделі в розпізнаванні патологічних станів печінки на зображеннях УЗД.

Загалом, в рамках дипломної роботи було успішно вирішено всі поставлені завдання. Розроблена модель нейромережі ResNet здатна ефективно класифікувати медичні зображення УЗД печінки, що відкриває можливості для автоматичної класифікації та підтримки діагностики цих зображень в клінічній практиці.

Отримані результати свідчать про великий потенціал використання штучних нейронних мереж для автоматизації та покращення діагностики хвороб печінки на основі медичних зображень УЗД. Продовженням дослідження може бути побудова більш складних моделей та розширення обсягу дослідження на інші типи медичних зображень та патології печінки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kang L., Ye P., Li Y., Doermann D. Convolutional neural networks for no-reference image quality assessment; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Columbus, OH, USA. 23–28 June 2014; pp. 1733–1740.
   URL: https://openaccess.thecvf.com/content_cvpr_2014/html/Kang_Convolutional_Neural_Networks_2014_CVPR_paper.html
2. Kim J., Lee S.H. Fully deep blind image quality predictor. IEEE J. Selected Topics in Signal Process. 2017;11:206–220. doi: 10.1109/JSTSP.2016.2639328. URl: https://yonsei.elsevierpure.com/en/publications/fully-deep-blind-image-quality-predictor
3. Bare B., Li K., Yan B. An accurate deep convolutional neural network model for no-reference image quality assessment; Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME); Hong Kong, China. 10–14 July 2017. URL: https://ieeexplore.ieee.org/abstract/document/8019508
4. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition; Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Las Vegas, NV, USA. 27–30 June 2016 URL: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf
5. Kim J., Lee S.H. Deep learning of human visual sensitivity in image quality assessment framework; Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Honolulu, HI, USA. 21–26 July 2017. URL:
   https://openaccess.thecvf.com/content_cvpr_2017/papers/Kim_Deep_Learning_of_CVPR_2017_paper.pdf
6. Gu J., Meng G., Wang L., Pan C. Learning deep vector regression model for no-reference image quality assessment; Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); New Orleans,

LA, USA. 5–9 March 2017. URL: https://link.springer.com/article/10.1007/s11263-020-01419-7

7. Liu X., van de Weijer D., Bagdanov D. RankIQA: Learning from rankings for no-reference image quality assessment; Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); Venice, Italy. 22–29 October 2017. URL: https://arxiv.org/abs/1707.08347
8. Bosse S., Maniry D., Müller K.-R., Wiegand T., Samek W. Deep neural networks for no-reference and full-reference image quality assessment. IEEE Trans. Image Process. 2018;27:206–219. doi: 10.1109/TIP.2017.2760518.
   URL: https://arxiv.org/abs/1612.01697
9. Ma K., Liu W., Zhang K., Duanmu Z., Wang Z., Zuo W. End-to-end blind image quality assessment using deep neural networks. IEEE Trans. Image URL: https://pubmed.ncbi.nlm.nih.gov/29220321/
10. Olaf Ronneberger, Philipp Fischer, Thomas Brox. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv. URL: https://arxiv.org/abs/1505.04597
11. Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. (2016). Densely Connected Convolutional Networks. arXiv. URL: https://arxiv.org/abs/1608.06993
12. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. (2015). Deep Residual Learning for Image Recognition. arXiv. URL: https://arxiv.org/abs/1512.03385
13. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. (2014). Going Deeper with Convolutions. arXiv. URL: https://arxiv.org/abs/1409.4842
14. Liu X, Faes L, Kale AU, et al. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. Lancet Digital Health. 2019;1(6):e271-e297. doi:10.1016/S2589-7500(19)30123-2 URL: https://www.thelancet.com/journals/landig/article/PIIS2589-7500(19)30123-2/fulltext

15. Yang B., et al. "Comparative Study of Deep Neural Network Architectures for Liver Ultrasound Image Classification." IEEE Transactions on Medical Imaging, vol. 38, no. 5, 2019. URL: https://www.sciencedirect.com/science/article/pii/S0933365722000963
16. Chen C., et al. "Transfer Learning with Convolutional Neural Networks for Liver Ultrasound Image Classification." Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2020. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5359213/
17. Lee D., et al. "Ensemble Deep Learning Models for Liver Ultrasound Image Classification." Journal of Medical Systems, vol. 44, no. 8, 2020. URL: https://www.mdpi.com/2306-5354/10/1/69
18. D. Ravi, V. Wong, and S. Lo, "Deep learning for healthcare: review, opportunities and challenges," Briefings in Bioinformatics, vol. 19, no. 6, pp. 1236-1246, 2018. URL: https://academic.oup.com/bib/article/19/6/1236/3800524
19. P. Saha, B. Koley, and S. K. Bandyopadhyay, "Application of ultrasound in medicine: A review," International Journal of Scientific Research in Science, Engineering and Technology, vol. 2, no. 1, pp. 30-37, 2016. URL: http://www.scielo.org.co/pdf/tecci/v8n15/v8n15a03.pdf
20. Chaudhuri, B. B., & Dey, N. (Eds.). (2020). Deep Learning Techniques for Biomedical and Health Informatics. Springer. https://dokumen.pub/deep-learning-techniques-for-biomedical-and-health-informatics-1st-ed-2020-978-3-030-33965-4-978-3-030-33966-1.html

    URL: https://arxiv.org/abs/1612.01697
21. "A Guide to Physical Examination and History Taking" (Багатоавторський посібник з фізичного обстеження та збору анамнезу) - автори Barbara Bates, Lynn S. Bickley. URL: https://www.amazon.com/Bates-Physical-Examination-History-Taking/dp/146989341X
22. "Atlas of Ultrasound Images of Liver Diseases" by Min Ren, Qinghai Peng, and Xin-Wu Cui. URL: https://liveratlas.org/

23. Gaillard F, Worsley C, Niknejad M, et al. Cirrhosis. Reference article, Radiopaedia.org (Accessed on 17 May 2023) https://radiopaedia.org/articles/cirrhosis

24. Baker, J.A.: Computer-aided detection (CAD) in screening mammography: sensitivity of commercial CAD systems for detecting architectural distortion. Am. J. Roentgenol 181(4), 1083–1088 (2003) https://link.springer.com/chapter/10.1007/978-3-030-35445-9_20

25. 1. Класифікатор диференціальної діагностики аутоімунного гепатиту та хвороби Вільсона на основі аналізу ультразвукових зображень печінки / Іванченко А. С., Бовсуновська К. С., Дикан І. М., Тарасюк Б. А., Павлов В. А., Настенко Є. А. // Біомедична інженерія і технологія. – 2021. – № 6. – С. 62-73. https://ela.kpi.ua/handle/123456789/45490

26. “Wavelet Decomposition Facilitates Training on Small Datasets for Medical Image Classification by Deep Learning” Histochem Cell Biol. 2021 Feb; 155(2): 309–317. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7957953/

27. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. 2015. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9767727/#B18

28. https://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/

Ресурси зображень

29. https://www.medicalnewstoday.com/articles/liver-ultrasound

30. https://www.mdpi.com/2075-1729/12/11/1877

31. https://www.researchgate.net/figure/Abdominal-ultrasound-demonstrating-a-liver-tumor-of-size-59-A-57-cm-in-hepatic-caudate_fig1_298796907

32. https://diagnoz.in.ua/diagnostika-ta-analizi/uzd/rozmiri-pechinki-v-normi-u-doroslikh-na-u/

33. https://svitppt.com.ua/medicina/hronichni-zahvoryuvannya-pechinki.html

34. https://www.mdpi.com/1424-8220/21/16/5304