

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
КАФЕДРА БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

«На правах рукопису»
УДК 616.1–616.7

До захисту допущено:

Завідувач кафедри

_____ Владислав ШЛИКОВ

«__» _____ 20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Медична інженерія»

зі спеціальності 163 «Біомедична інженерія»

на тему: «Комплекс на основі машинного навчання для визначення динаміки температурних градієнтів поверхні серця і коронарних судин»

Виконала:

студентка II курсу, групи БМ-91мп
Шапошник Ольга Геннадіївна



Керівник:

в.о. зав.каф. БМІ, д.т.н, доцент,
Шликов Владислав Валентинович

Рецензент:

Доцент, к.т.н,
Антонова-Рафі Юлія Валеріївна

Нормоконтролер:

Інженер 1 категорії,
Андреев Петро Іванович

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет біомедичної інженерії
Кафедра біомедичної інженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність – 163 «Біомедична інженерія»

Освітньо-професійна програма «Медична інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Владислав, ШЛИКОВ

« ___ » _____ 20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Шапошник Ольга Геннадіївна

1. Тема дисертації «Комплекс на основі машинного навчання для визначення динаміки температурних градієнтів поверхні серця і коронарних судин», науковий керівник дисертації Шликов Владислав Валентинович д.т.н, доцент, затверджені наказом по університету від «11» листопада 2020 р. № 3282
2. Термін подання студентом дисертації 11.12.2020
3. Об'єкт дослідження: градієнт температури серця на термограмах.
4. Вихідні дані: термограми серця під час відкритих операції на ньому в моменти охолодження та нагрівання.
5. Перелік завдань, які потрібно розробити: Порівняти декілька методів класифікації термознімків та обрати оптимальний для виявлення градієнта температури, який вищий за 3 градуси. Розробити спосіб класифікації термознімків серця для виявлення певних патологій, коли градієнт температури більше за 3 градуси Цельсія, використовуючи сучасні методи машинного навчання. Розробити систему динамічного аналізу на основі нейронної мережі та визначити показники результативності проведення операції на відкритому серці. Спроекувати та розробити інтерфейс цієї системи, який забезпечить зручне та інтуїтивно зрозуміле користування. Проаналізувати методи покращення алгоритму класифікації нейронної мережі.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: діаграми та блок-схеми програмного додатку, скріншоти роботи програми, презентація до захисту роботи

7. Орієнтовний перелік публікацій: «Нейронна мережа для аналізу термограм під час відкритої операції на серці» у журналі "Біомедична інженерія і технологія".

8. Дата видачі завдання: 26.10.2020

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Визначення теми наукового пошуку	Січень 2020	
2	Огляд літератури та патентний пошук	Січень-лютий 2020	
3	Аналіз можливих методів класифікації зображень	Лютий-березень 2020	
4	Вибір оптимального методу класифікації	Березень 2020	
5	Реалізація обраного методу	Квітень 2020	
6	Пошук аналогів для обраного методу класифікації	Травень 2020	
7	Реалізація методів-аналогів	Червень-липень 2020	
8	Порівняння отриманих результатів та вибір базового методу аналізу	Серпень 2020	
9	Препроцесінг термознімків	Вересень 2020	
10	Проектування програмного продукту	Вересень-жовтень 2020	
11	Реалізація програмного продукту	Жовтень-листопад 2020	
12	Тестування роботи програми	Листопад 2020	
13	Оформлення роботи	Листопад-грудень 2020	
14	Підготовка до захисту	Грудень 2020	
15	Захист магістерської дисертації	21.12.2020	

Студент

Ольга ШАПОШНИК

Науковий керівник

Владислав ШЛИКОВ

РЕФЕРАТ

Тема магістерської дисертації: «Комплекс на основі машинного навчання для визначення динаміки температурних градієнтів поверхні серця і коронарних судин».

Обсяг роботи становить 104 сторінок, міститься 33 ілюстрацій, 11 формул, 15 таблиць. Загалом опрацьовано 52 джерела.

Мета: Система на основі машинного навчання для автоматизації аналізу термограм в динаміці, яка дає можливість оцінити успішність проведення операції на відкритому серці.

Задачі:

1. Порівняти декілька методів класифікації термознімків та обрати оптимальний для виявлення градієнта температури, який вищий за 3 градуси.
2. Розробити спосіб класифікації термознімків серця для виявлення певних патологій, коли градієнт температури більше за 3 градуси Цельсія, використовуючи сучасні методи машинного навчання.
3. Розробити систему динамічного аналізу на основі нейронної мережі та визначити показники результативності проведення операції на відкритому серці.
4. Спроекувати та розробити інтерфейс цієї системи, який забезпечить зручне та інтуїтивно зрозуміле користування.
5. Проаналізувати методи покращення алгоритму класифікації нейронної мережі.

Основні результати: було здійснено аналіз сучасних методів машинного навчання та на їх основі побудовано класифікатор для визначення градієнтів температури серця з точністю 65% та на цій основі розроблено програмний додаток для динамічного аналізу термограм з можливістю надати кількісну характеристику успішності операції з ішемічними ушкодженнями серця.

Ключові слова: нейронна мережа, термографія, машинне навчання, аналіз зображень, Python.

ABSTRACT

The topic of the Master thesis is: " A machine learning-based complex for determining temperature gradients dynamics of the surface of the heart and coronary arteries."

The size of the report is 104 pages, contains 33 illustrations, 11 formulas, 15 tables. In total, there have been processed 52 sources.

Objective: Machine-based system for automation of thermogram analysis in dynamics, which makes it possible to assess the success of open heart surgery.

Task:

1. Compare several methods of classification of thermal images and choose the optimal temperature gradient that is higher than 3 degrees.

2. To develop a method of classification of thermal images of the heart to detect certain pathologies when the temperature gradient is more than 3 degrees Celsius, using modern methods of machine learning.

3. Develop a system of dynamic analysis based on the neural network and determine the effectiveness of open heart surgery.

4. Design and develop the interface of this system, which will provide convenient and intuitive use.

5. Analyze methods for improving the neural network classification algorithm.

Main results: analysis of modern machine learning methods was performed and on their basis a classifier for determining heart temperature gradients with an accuracy of 65% was built and on this basis a software application for dynamic analysis of thermograms was developed with the ability to quantify the success of ischemic heart surgery.

Keywords: neural network, thermography, machine learning, image analysis, Python.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ТЕОРЕТИЧНА ЧАСТИНА	10
1.1 Небезпека серцево-судинних захворювань	10
1.2 Проведення операцій на серці	11
1.3 Методи проведення операцій на відкритому серці.....	13
1.4 Розповсюдженість використання термограм в медицині.....	15
1.5 Використання термограм під час операцій на серці	22
1.6 Автоматизація обробки термограмів	23
Висновки до розділу 1	26
РОЗДІЛ 2 МАТЕРІАЛИ ТА МЕТОДИ	27
2.1 Матеріали.....	27
2.2 Методи.....	31
Висновки до розділу 2	35
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА	36
3.1 Розробка нейронної мережі для аналізу термограм	36
3.2 Тестування та аналіз класифікаторів	39
3.3 Нечітка математика.....	41
3.4 Аналіз гістограм	44
3.5 Методи покращення алгоритму класифікації	48
Висновки до розділу 3	50
РОЗДІЛ 4 ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ	51
4.1 Контекстна діаграма IDEF0	52
4.2 Діаграма декомпозиції IDEF0	53
4.3 Діаграма IDEF3.....	54
4.4 Дерево вузлів	54
4.5 Діаграма класів	55
4.6 Use-case діаграма.....	56
4.7 Діаграма послідовності.....	57

4.8 Діаграма кооперацій	58
4.9 Діаграма станів	59
Висновки до розділу 4	60
РОЗДІЛ 5 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	61
5.1 Вибір мови програмування	61
5.2 Вибір середовища розробки.....	67
5.3 Порівняння бібліотек для машинного навчання.....	69
5.4 Опис основних бібліотек.....	74
5.5 Модель життєвого циклу.....	76
5.6 Розробка інтерфейсу	83
5.7 Вирішення практичної задачі	86
Висновки до розділу 5	90
РОЗДІЛ 6 РОЗРОБКА СТАРТАП ПРОЕКТУ	91
6.1 Опис ідеї стартап проекту	91
6.2 Технологічний аудит проекту.....	92
6.3 Аналіз ринкових можливостей запуску стартап-проекту.....	92
6.4 Розроблення ринкової стратегії проекту	96
6.5 Розроблення маркетингової програми стартап-проекту	96
Висновок до розділу 6	98
ВИСНОВКИ.....	99
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	100

ВСТУП

На сьогодні термограми отримали велику популярність, як метод скрінінгу та діагностики різних патологій на ранніх стадіях, його використовують для виявлення злоякісних пухлин, для дослідження запалювальних процесів, для виявлення захворювань серцево-судинної системи, а також як допоміжний засіб для моніторингу стану серця при проведенні відкритих операції на ньому. Але обробка та аналіз термограм носять приватний характер, підходи до аналізу найчастіше залежать від досліджуваного органу, а більшість діагнозів ставляться по візуальному аналізу, що є суб'єктивною оцінкою і може вносити похибки при винесенні діагнозу без належного досвіду[1].

Температура тіла залежить від безлічі внутрішніх та зовнішніх факторів. Температура не є абсолютним показником. Тому оцінювати результати отриманої термограми потрібно індивідуально, звертаючи більше уваги не на абсолютні показники, а на різницю температур[2]. У деяких випадках достовірність результатів аналізу термографії лікарем становить всього 60%, і такі низькі результати пов'язані з помилковим розшифруванням термограм. Саме цей метод дає велику кількість хибнопозитивних результатів і призводить до гіпердіагностики. І саме це головна причина, по якій поки термограма не мають великої популярності в якості скрінінгового обстеження.

За допомогою термографії можна визначити різницю температур з точністю до $0,08^{\circ}\text{C}$, тому сам по собі метод вважається досить точним. Зараз розроблено багато методів автоматичного аналізу в різних сферах медицини. Для прогнозування кардіоваскулярних відхилень розроблений тест на васкулярну реактивність. Різні методи машинного навчання зазвичай використовують для виявлення ракових пухлин в щитовидній та молочних залозах за допомогою термознімків[3]. Найпопулярнішими методами аналізу залишаються різноманітні варіанти нейронних мереж та текстурний аналіз[4].

Головна проблематика при аналізі термограм – для кожної окремої області потрібне створення своїх, специфічних методів аналізу зображень. Навіть уже

розроблені алгоритми, вимагають коригування для застосування їх в аналізі в інших сферах, в поточному випадку, для термограм серця[5].

Наукова новизна роботи полягає у розробці системи на основі машинного навчання для динамічного аналізу термограм серця. Зазвичай аналізуються окремі знімки, але не їх впорядковані в часі масиви, які здатні показати загальну, більш повну картину стана серця, на основі якої можна робити висновки про успішність проведеної операції на відкритому серці.

Тема: Комплекс на основі машинного навчання для визначення патологічних станів серця

Мета: Система на основі машинного навчання для автоматизації аналізу термограм в динаміці, яка дає можливість оцінити успішність проведення операції на відкритому серці.

Задачі:

1. Порівняти декілька методів класифікації термознімків та обрати оптимальний для виявлення градієнта температури, який вищий за 3 градуси.
2. Розробити спосіб класифікації термознімків серця для виявлення певних патологій, коли градієнт температури більше за 3 градуси Цельсія, використовуючи сучасні методи машинного навчання.
3. Розробити систему динамічного аналізу на основі нейронної мережі та визначити показники результативності проведення операції на відкритому серці.
4. Спроекувати та розробити інтерфейс цієї системи, який забезпечить зручне та інтуїтивно зрозуміле користування.
5. Проаналізувати методи покращення алгоритму класифікації нейронної мережі.

РОЗДІЛ 1

ТЕОРЕТИЧНА ЧАСТИНА

1.1 Небезпека серцево-судинних захворювань

На сьогоднішній день серцево-судинні захворювання є самою популярною причиною смерті в світі та становить 31% від загального числа смертей. За даними 2016 року, від серцево-судинних захворювань померло біля 17,9 мільйонів людей, з них 85% через інсульт або приступ серця та біля 75% випадків приходить на країни з низьким або середнім рівнем заробітку[6].

При цьому більшість випадків серцево-судинних захворювань можна попередити. Обтяжуючими причинами та факторами ризику є нездорове харчування, надмірна вага, куріння, алкоголь. Але в групу ризику також включаються люди з підвищеним кров'яним тиском, діабетом, гіперліпідемією та іншим. Їм потрібно ретельніше моніторити свій стан здоров'я, аби мати змогу виявлення захворювань зв'язаних з серцево-судинною системою на початковому етапі, коли можливе медикаментозне або консультативне лікування.

Серцево-судинні захворювання – це група захворювань, яка об'єднує захворювання серця та кровоносних судин, до них можна віднести наступні групи захворювань[7].

Ішемічна хвороба серця – захворювання, яке пов'язане з втратою можливості коронарних судин забезпечувати необхідною кількістю крові серце.

Захворювання судин головного мозку – захворювання, при якому судини мозку перестають адекватно забезпечувати мозок кров'ю

Захворювання периферичних артерій – захворювання кровоносних судин, які забезпечують кровопостачання до верхніх та нижніх кінцівок.

Ревмокардит – ураження м'язів серця чи клапанів в результаті ревматичної атаки через стрептококові бактерії.

Вроджений порок серця – деформація в будові серця з самого народження.

Інфаркти чи інсульти є гострими захворюваннями, які характеризуються закупоркою судів тромбами, які перешкоджають проходженню потоку крові до серця чи мозку. Накопичення жирової тканини на внутрішній стінці кровоносних судин, які забезпечують кров'ю мозок чи серце – найрозповсюдженіша причина цього захворювання. Також можливим наслідком може бути кровотечі з судин в мозок. В основному, причина інфаркта чи інсульта є ряд факторів ризику, такі як нездорові звички, незбалансоване харчування, зайва вага, відсутність фізичних вправ, постійно підвищений тиск та ряд хронічних захворювань, такі як, діабет, гиперліпідемія[7, 8].

Окрім цього, серцево-судинні захворювання є відображенням основних культурних, економічних та соціальних змін в суспільстві. Таким чином, більш ніж в 75% випадків смерті від серцево-судинних захворювань припадає на країни з низьким та середнім доходом.

Люди в таких країнах часто не мають можливості користуватися сучасними програмами першої медичної допомоги для ранньої діагностики та виявлення факторів ризику, які можуть призвести до летальних наслідків. Також такі люди з країн з низьким та середнім доходом, які страждають від захворювань серцево-судинної системи, не мають належного доступу до ефективних методів лікування, які б підходили під їх індивідуальні випадки. Через це, багата людей помирають від серцево-судинних та інших неінфекційних захворювань в молодими, в час своїх самих продуктивних років[9].

Іншим важливим з цього наслідком є те, що в подібного рівня країнах, люди з такими захворюваннями стають ще бідніші через великі витрати на медицину, через що з кожним роком їм стає все складніше забезпечувати себе можливістю в лікуванні.

1.2 Проведення операцій на серці

Різні захворювання серцево-судинної системи вимагають різних методів лікування, від медикаментозного до великих і складних операцій з подальшим

дотриманням профілактичних мір, які здатні до 75% попередити повторний рецидив. На ранніх стадіях, деякі захворювання можливо лікувати медикаментозно, але в певних випадках не обійтися без операції. Це може бути аортокоронарне шунтування, балона ангіопластика, пересадка серця, заміна клапана чи вживлення штучного серця[10].

Розрізняють декілька видів операцій на серці, вони можуть класифікуватися по виду операцій та по її терміновості.

Екстрені операції проводять одразу після постановки діагнозу через те, що будь-які затримки можуть привести до летальних наслідків.

Під час невідкладних операцій ситуація стан пацієнта теж вважається небезпечною, але при цьому терміновості в операції немає. У пацієнта є можливість почекати пору днів та підготуватися до операції.

Планові операції назначаються, коли пацієнт має стабільне задовільне самопочуття і день операції обирається разом з лікарем, але подібні операції те мають певну межу, далі якої їх відкладати стає небезпечно.

Також операції на серці можна поділити на закриті, відкриті та рентгенохірургічні[11].

Під час закритих операцій саме серце не чіпається, вони виконуються поза нього та не потребують додаткового оснащення. Часто їх використовують як початковий етап втручання.

Рентгенохірургічні операції є доволі молодим методом лікування, але він отримав велику популярність через мінімальне втручання, адже операції проводяться спеціальними катетарами, які вводять в полость серця через вени чи артерії. Зараз цей метод може використовуватися як основний чи додатковий, разом зі звичайною операцією, бо він є дієвим тільки для ряду захворювань.

Відкриті операції – вид операції на серці, при якому необхідно відкрити полость серця та провести усунути патології безпосередньо на ньому. Для проведення такою операції необхідно використовувати додаткові технічні апарати, такі як апарат штучного кровообігу. Під час проведення операції необхідно зупинити серце, при неможна залишати організм без кровопостачання. Для цього

серце підключають до подібного апарату. Венозна кров пацієнта проходить через апарат, де відбувається газообмін, кров віддає вуглекислий газ та насичається киснем, як це відбувається в легких людини, в малому колі кровообігу. Далі насичену киснем, вже артеріальну кров запускають в аорту та вона йде по всьому тілу.

Такі технології дозволяють значно полегшити хірургічне втручання та дозволяють хірургу, без великих ризиків, проводити складні багатогодинні операції[12].

1.3 Методи проведення операцій на відкритому серці

Як було зазначено раніше, при відкритих операціях на серці, необхідно його зупинити, бо всі втручання проводяться на зупиненому серці, тому що інакше провести операцію було б неможливо.

Перші методи зупинки серця здійснювалися за допомогою штучного охолодження тіла людини до температури 30 градусів Цельсія, що дозволяло зупинити серце на невеликий проміжок час, від 15 до 20 хвилин, чого було достатньо для проведення нескладної операції. При такому підході, серце зупинялося через те, що організм перестав вимагати великої кількості кисню, для підтримки працездатності, через що і кровоток починав ставати повільнішим, і серце працювати менш інтенсивно[12].

Цей метод складався з того, що після того як пацієнта введуть під наркоз, до в коло нього починали складати спеціальні пакети з дрібним льодом, при цьому лікарі мали уважно стежити за датчиками тиску та температури. Після того, як температура падала до 32 градусів Цельсія, лід забирали, але температура продовжувала за інерцією спадати ще на 2-3 градуси. Таким чином, лікар мав 15-20 хвилин для проведення операції, інакше у пацієнта може початися гіпоксія, що призведе до незворотних наслідків в організмі. Після проведення операції, серце запускали знов, за допомогою спеціальних препаратів або стимуляцій та тіло починали зігрівати за допомогою теплого повітря.

Такий метод довгий час був основним під час проведення операцій на серці, він був не дорогим та давав можливість добитися гарних результатів. Але основна проблема так і лишалася в тому, що час проведення операцій був занадто обмежений.

Цю проблему змогли вирішити за допомогою винаходження апарату штучного кровообігу, що дозволяло повністю ізолювати серце під час операції, при цьому весь інший організм продовжував працювати в звичайних умовах.

Апарат штучного кровообігу складається з фізіологічного та механічного блоків. До фізіологічного блоку відносяться оксигенатор, який здійснює функцію газообміну і насичує кров киснем, при цьому забираючи з неї вуглекислий газ, та артеріальний насос, який і виконує основну функцію серця. Механічний блок, представляє собою блок насосів і пристрої для постійного моніторингу стану пацієнта та датчиками управління. На сьогоднішній день, всі частини апарату штучного кровообігу, які контактують з кров'ю пацієнта, є одноразові та індивідуальні для кожного пацієнта[6].

Але навіть при цьому існує декілька нюансів для роботи. Для початку необхідно забезпечити, аби кров не зверталася, для цього кров пацієнта завчасно розріджують, вводячи гепарин чи його аналоги. Наступним етапом, необхідно підключити людину до апарату штучного кровообігу. Перш за все, трубки фізіологічного блоку з'єднуються з судинами пацієнта і при цьому виникає подвійний кровообіг, бо частина крові продовжує йти через серце, а частина починає рухатися по трубкам апарату штучного кровообігу.

Потім сосуди щільно притискаються до трубок і кров починає рухатися тільки по апарату штучного кровообігу. Венозна кров поступає в оксигенатор, насичується киснем та віддає вуглекислий газ, проходить стадію фільтрації від дрібних бульбашок повітря і потім наноси виштовхують вже артеріальну кров в аорту. Також важливою частиною є наявність теплообмінника, який забезпечує стабільну температуру крові людини і не дає їх охолонути.

Таким чином здійснюється зупинки серця, при цьому кровообіг продовжує функціонувати та виконувати свої функції. Хоча навіть при такому методі, серце

продовжує скорочуватися, але відбувається це досить слабо, що не заважає хірургу. Повністю зупинити серце не є можливим, адже воно має свою власну систему коронарних судин, які необхідні для підтримки його життєздатності.

Наразі продовжуються модернізуватися та покращуватися системи та протоколи проведення операцій на відкритому серці. Наприклад, одним з методів, який допомагає попередити кисневу недостатність в серці, базується на консервації серця. Такий метод має назву холодова кардіоплегія.

Перед операцією пацієнту необхідно ввести спеціальний розчин прямо в аорту. Цей кардіоплегічний розчин складається з індивідуально підбраного електролітичного складу з великою часткою калія, який має бути температурою не вище чотирьох градусів. Це змушує серце швидко зупинитися приблизно на 20 хвилин, що дозволяє хірургу проводити операцію більш точно та швидше.

При цьому, завдяки складу розчину, серце захищено від гіпоксії, та забезпечує збереження енергетичних запасів серця навіть при довгострокових операціях.

Також під час операції на серці використовується і техніка для моніторингу стану пацієнта. Одним з важливих факторів в цьому є температура тіла та самого серця. Зараз велику популярність не тільки в кардіохірургії, а і в медицині в цілому, отримали методи безконтактних вимірів температури. В цьому випадку це важливо, аби мати змогу попередити різкі спади чи підйоми під час нагрівання чи охолодження серця, адже, якщо градієнт температур вищий за 5 градусів Цельсія, то подібна ситуація може бути небезпечною для тканин міокарда та призвести до рубця та відмирання тканини.

1.4 Розповсюдженість використання термограм в медицині

Наразі в медичній сфері все більше починають застосовуватися неінвазійні методи діагностики, які є більш безпечні для пацієнтів та лікарів, завдяки відсутності контакту людей з біологічними речовинами, збереження цілісності тканин та спрощення проведення огляду. Одним з таких методів є термографія, яка

вже довела свою результативність та абсолютну безпечність під час використання[13].

Медична термографія є методом дослідження та моніторингу стану пацієнта за допомогою теплового випромінювання, яке йде від тіла. Тепловізор або термограф, працюючи в інфрачервоному випромінюванні, яке є абсолютно безпечним та немає ніяких протипоказань для використання на людях, може сприймати теплове випромінювання від пацієнта та перетворювати його в зображення, які називають термограмами. На цих знімках зареєстрований розподіл тепла конкретної зони тіла людини, також таким чином будуються температурні карти пацієнта, яка також є базовою можливістю для оцінки температури об'єкта[14].

При цьому, його температурна чутливість при 30 градусів Цельсія досягає 0,02 градусів, це означає, що при замірюванні температури об'єктів, температура яких становить 30 градусів, можна досягнути точності вимірів в точності до сотих.

Сам термограф має нескладну будову (рис. 1.1).

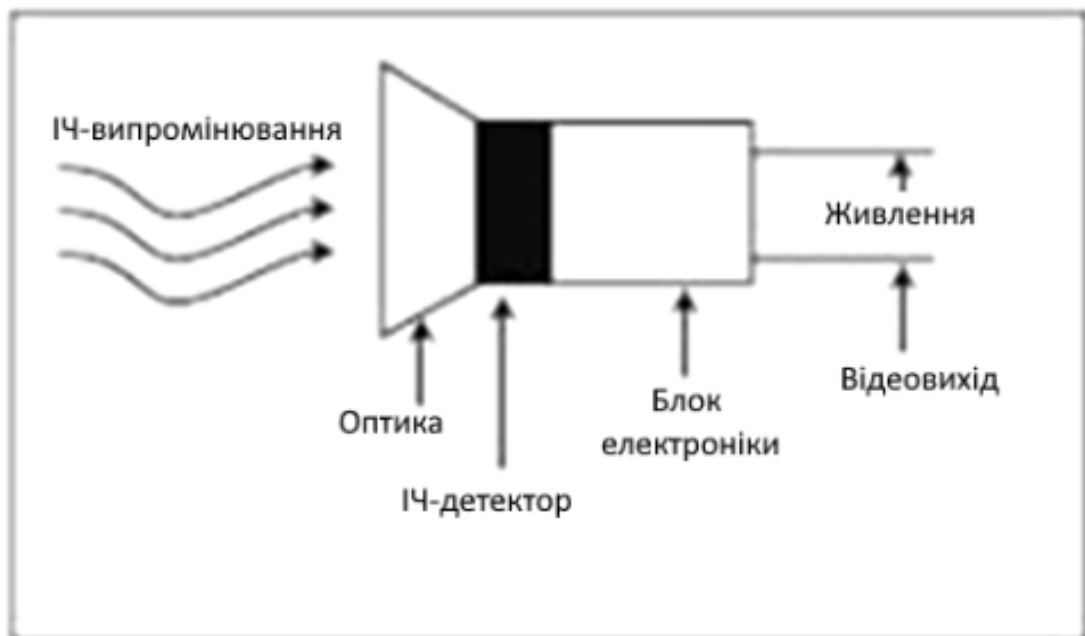


Рисунок 1.1 – Спрощена схема тепловізора

Основним його елементом є фоточутлива матриця та германієвий об'єктив. Інфрачервоне випромінювання через германієвий об'єктив потрапляє на детектор,

чутливі елементи якого змінюють властивості, що призводить до зміни вхідного сигналу (рис.1.2), який потім обробляється блоком електроніки, інформація про теплове випромінювання видозмінюється та подається на вихідну матрицю у вигляді термознімка (рис.1.3), які в свою чергу можуть бути монохромні чи кольорові[15].

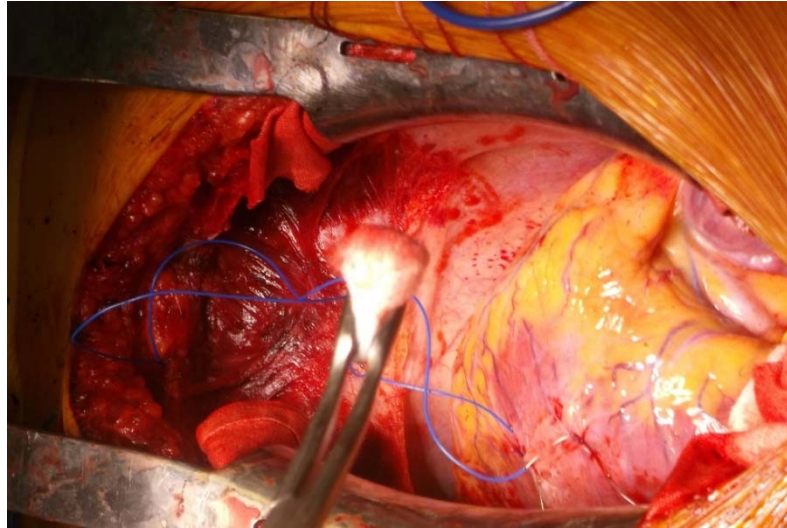


Рисунок 1.2 – Зображення серце під час операції

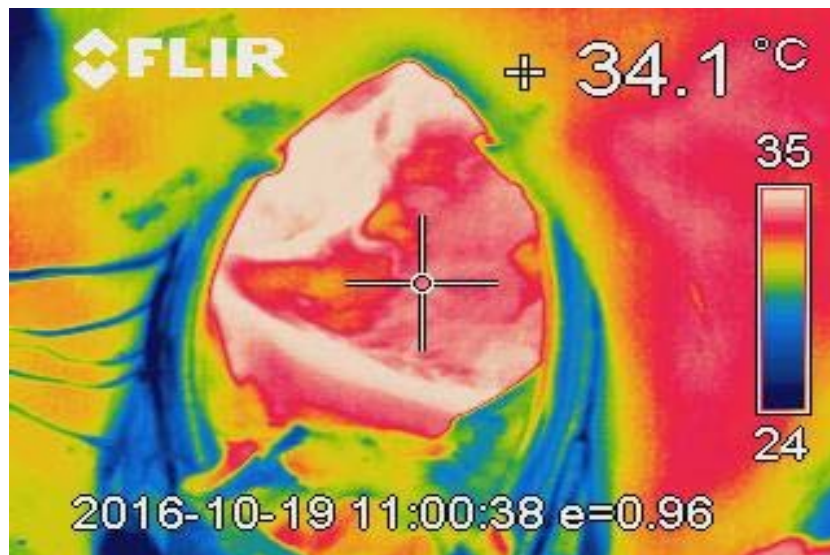


Рисунок 1.3 – Зображення серця в інфрачервоному спектрі через термограф.

За допомогою теплових карт тіла, які отримуються за допомогою термографа, можна відслідковувати різні види патологій та відхилень. Таким чином, на сьогоднішній день такі технології знаходять застосування для

діагностики онкологічних захворювань, запалень, внутрішніх ушкоджень та багато іншого. Під час подібної діагностики, знімається теплова карта необхідної ділянки тіла і, на основі розподілу температури, лікарями робиться висновок про наявності чи відсутності патологічних відхилень, які можна спостерігати, зазвичай, як області різкої зміни градієнта температури[16].

Сучасні технології дозволяють отримати зображення розмірністю до 640x480, при цьому точність може доходити до тисячних, в районі від 0,007 до 0,015 градусів Цельсія. Така висока точність можлива через використання матричних камер інфрачервоного спектру замість камер з одноелементними інфрачервоними фотоприймачами. Це обумовлено тим, що реєстрація інфрачервоного зображення відбувається одночасно кожним з елементів матриці приймача, що і дозволяє суттєво збільшити час інтеграції інфрачервоного випромінювання за один кадр.

Також, на даний момент, вирішені проблеми пов'язані з вартістю подібної апаратної техніки. З'являється все більше компактних пристроїв, які за своєю вартістю не перевищують ціни в 200\$, при цьому зберігаючи задовільні характеристики: спектральний діапазон 8-12 мкм, чутливість – 0.03-0.04 градусів Цельсія, частота кадрів – 25 в секунду. Подібних характеристик вдосталь, аби мати змогу використовувати пристрій в медичних цілях, для скрінінгу та діагностики базових захворювань[17].

Також, такі камери можуть слугувати для використання в портативних цілях, наприклад, для використання в машинах швидкої допомоги чи амбулаторних закладах. При цьому, простота проведення процедури не потребує додатково оснащених приміщень чи додаткової техніки. Подібні маніпуляції можна проводити в домашніх умовах, але при дотриманні температурних режимів довколишнього середовища, яке може додати на зображення додаткові артефакти.

Багато статей та дослідників описують перспективність та актуальність використання термографії, як засіб діагностики при запальних процесах, захворюваннях серцево-судинної системи, виявлення онкології на ранніх етапах.

Але при цьому подібним напрямкам приділяють недостатньо уваги, особливо при діагностиці серцево-судинної системи.

В своїх дослідках науковці опирають на використання вже застарілих технологій, а саме термографів старого покоління, які не мали достатніх можливостей. І методичні рекомендації по їх використанню в діагностиці теж не відповідають сучасності, що не дозволяє створити повноцінні медичні методики застосування реєстрації інфрачервоного випромінювання для пацієнтів. Хоча сам метод подібної діагностики має велику базу та доказову достовірність[17].

Термографія отримала широке використання для діагностики онкологічних процесів в організмі, її часто використовують для діагностики ракових пухлин в молочних залозах, щитовидної залози, лімфатичних вузлах та іншим частинах тіла. Відомо, що тканини, в яких спостерігається онкологічні та пухлинні тканини мають більш високу температуру ніж здорові тканини, які її оточують, тому що такі тканини мають сильнішу проліферальну активність та інтенсивний метаболізм, на що і спираються лікарі при діагностиці онкологічних захворювань за допомогою інфрачервоного випромінювання.

Окрім самої діагностики наявності пухлинних процесів в організмі, також з певною точністю можна відрізнити доброякісні пухлини від злоякісних. Зазвичай, якщо пухлина є злоякісною, то її температура вище на 1.5 – 3 градуси Цельсія від оточуючих тканин, а якщо пухлина доброякісна, то її температура на 1.4 – 2.5 градусів Цельсія нижча за оточуючі тканини.

Також, значною перевагою подібного методу те, що під час пухлинних процесів змінюється і кровоток біля цієї частини тіла, через що і на поверхні тіла можна спостерігати внутрішні зміни температури, що і дає змогу застосовувати інфрачервоні датчики для діагностичних цілей, які на відміну від рентгенівського дослідження, яким зараз зазвичай використовують для подібних діагностичних цілей, є повністю безпечним та крім цього дає змогу бачити зміни в структурі тканини на ранніх етапах, чого зробити рентгенівське дослідження не в силах.

Окрім діагностик онкології, термографія отримала популярність і для діагностики серцево-судинної системи, це легко пояснити тим, що кровоносні

судини є терморегулюючим чинником в організмі, і тому використання для них методів, які ґрунтуються на виявленні теплового випромінювання, є самим актуальним[18].

Таким чином, якщо на температурній карті тіла можна побачити зони з пониженою температурою в районі нижніх та верхніх кінцівок, це може свідчити про проблеми в кровопостачанні та його недостатності. Подібні діагностики показують себе як дієвий метод для виявлення на ранніх стадіях варикозного розширення вен як нижніх так і верхніх кінцівок. В нормі, температура шкіри в області вен, має середнє значення в порівнянні з прилеглими до неї тканинами, але при наявності варикозного захворювання, можна спостерігати картину, що температурне значення в цих областях має підвищене значення на 1 – 2 градуси Цельсія, в порівнянні від оточуючих ділянок.

Також за допомогою різниці температур, можна встановлювати не лише наявність захворювання, а і на якій стадії воно знаходиться. Наразі існують багато методів діагностики проблем судин не тільки які вже наявні, а існують способи будувати прогнози по розбиту захворювань та ускладнень, зокрема для прогнозування ішемії кінцівок.

Також під час різних клінічних досліджень було визначено, що за допомогою термографії можна виявляти відхилення в порушеннях хребта таких як: остеопороз, астеоартроз, метастази тіл хребців та інші. При подібному роду захворювань на термокартах можна спостерігати зміну градієнта температури від 0.8 до 3 градусів Цельсія. Тут багато чого залежить від індивідуальних особливостей пацієнта та від конкретного виду захворювання.

В нормі температура спини має бути однорідною та її різниця не має перевищувати 0.2 градуси Цельсія, інакше будь-які інші отримані значення можна розцінювати як наявність патологій. Таким чином, метод термографії доцільно використовувати для додаткової діагностики та для попередньої діагностики перед тим як використовувати традиційні методи, аби мати змогу розуміти більш точно представлення про явну наявність захворювань і використовувати локальні методи, саме на конкретну ділянку тіла[19].

Окрім цього, подібні методи скрінінгу можна успішно застосовувати для відстеження результативності проходження реабілітації. Таким чином ряд досліджень показали, що під час реабілітації поперекового відділу при остеохондрозі, при успішній реабілітації можна спостерігати, як покращувався стан пацієнта після реабілітації. У контрольній групі можна було спостерігати зниження температури на 2 – 3 градуси. Також подібні експерименти ставилися і для пацієнтів з наявною плоскостопістю, після курсу реабілітації їх стан ставав стабільнішим та можна було спостерігати за відновленням кровопостачання до стоп та підвищенням температури в цих зонах на 2 – 4 градуси Цельсія, що свідчить про нормалізацію кровопостачання.

Як було зазначено раніше, при наявному гарному кровообігу біля зони цікавості, можна прослідкувати за їх зміною за допомогою термографії. Отже, опираючись на ці знання, можна висунути припущення, що подібну діагностику можна проводити і для щитовидної залози.

Вона оточена великим кровотоком, який бере активну участь під час обміну речовин. Тому температура тіла над нею швидко змінюється в залежності від наявності додаткових факторів та наявності патологій. Якщо розглядати термограму здорової людини, то на ній не можливо розрізнити наявності якихось конкретних контурів чи артефактів в місці розташування щитовидної залози, лише однорідну структуру.

Експериментально було виявлено, що з віком на щитовидній залозі можна спостерігати легкі прояви гіпотермії, що свідчить про наявність незворотних, вікових процесів, таких як інволюція залози. При гіперплазії на знімках фіксується наявність гіпертермії, це передумовлено тим, що залоза втрачає свої функціональні властивості, що і призводить до зниження температури[20].

Через те, що в нормі щитовидна залоза має однорідний вигляд на термознімках, це полегшує можливість діагностики багатьох захворювань, особливо раку та аденомів. При онкологічних захворюваннях спостерігається яскраво виражена область зв значно підвищеною температурою, по аналогії з молочними залозами.

Підсумовуючи все вищесказане та результати попередніх досліджень в медицині за допомогою інфрачервоного випромінювання, можна сказати, що цей метод активно розвивається та знаходяться все нові сфери його використання, що надає актуальність будь-якій роботі з термограмами в медицині[21].

Подібна методика є повністю безпечна для людини та немає ніяких обмежень чи протипоказань. Також вона неінвазивна та не потребує великих витрат для її проведення, що полегшує можливість її впровадження в більшій кількості медичних закладів, при цьому не закупляючи дороге обладнання чи відводячи спеціальні приміщення для апаратури. Вона може використовуватися в багатьох сферах, як основний інструмент діагностики, додатковий чи як додаткове обладнання скрінінгу етапів лікування.

Головним суттєвим мінусом є те, що для надання популярності та розповсюдженості цього методу необхідно зібрати велику прикладну, доказову базу даних не тільки для кожного виду органу, який можливо обстежувати, а і для кожного захворювання окремо, навіть якщо вони мають спільний характер чи аналоги в інших органах. Також через те, що термограму, зазвичай, розшифровують лікарі, які мають змогу спиратися тільки на свій власний та знання, часто вони неправильно ставлять результат обстеження. На температуру тіла людини може впливати безліч факторів, при тому і зовнішніх, які здатні спотворити результати, тому необхідно впровадження автоматичних систем діагностики термознімків, які здатні працювати з мінімальною похибкою при постановці діагнозів[22].

1.5 Використання термограм під час операцій на серці

Як було зазначено раніше, при відкритих операціях на серці є важливим стеження за його температурою та моніторинг небезпечних станів, коли охолодження чи нагрівання відбуваються занадто швидко. Термографія в таких випадках є найкращим рішенням. Вона дозволяє бачити широку картину стану

органу, на відміну від класичних методів виміру температури, коли є можливість проводити виміри в один момент часу лише на одній, невеликій ділянці.

Різка зміна температури може призвести до відмирання тканин та утворення рубця. Також при ішемічній хворобі такі дані можуть ставати маркерами для виявлення локалізації проблем з коронарними судинами. Опираючись на те, що ішемічна хвороба пов'язана з порушенням кровопостачання в коронарних артеріях, можна сказати, що після операції таких зон, з різкими скачками температури має стати менше, або зникнути взагалі[23].

Різниця температур при гіпертермії становить 3 – 4 градуси, а при ішемічній хворобі не має перевищувати 7 градусів Цельсія. Також експериментальні дані показують, що при наявності хронічних запалень можлива наявність різниці температур в 0.7 – 1 градусів та при гострому або деструктивному процесі в міокарді становить 1.5 – 2 градуси Цельсія.

1.6 Автоматизація обробки теомознімків

Одним із актуальних видів задач на класифікацію є задачі з медичної сфери. Розробка систем, які здатні автоматично аналізувати та класифікувати інформацію на основі математичних методів, машинного навчання та штучного інтелекту, спираючись на медичне підґрунтя, є актуальною та затребувана технологія. Подібне алгоритмічне встановлення діагнозів, забезпечує більш точні результати та менше помилок першого та другого роду, що дозволяє значно полегшити роботу лікарю, що значно покращить роботи медичної сфери[24].

Але основні проблеми при розробці подібних систем виникають не тільки через недостатність матеріалів для аналізу і проблеми з використанням медичних даних пацієнта, а ще і через недостатню структурування методів постановки діагнозів, які є найбільш значущі за признаками при постановці діагнозу. Більшість подібних даних є частково суб'єктивними та залежать від майстерності та обізнаності лікаря, який ставить діагноз.

Більшість подібних проблем можна вирішити за допомогою створення апаратно-програмного комплексу, який буде включати в себе базу даних захворювань, методів їх діагностики та програмних, автоматизованих рішень класифікації. Але не дивлячись на те, що зараз рівень розвитку обчислювальної потужності та математичних методів, на яких ґрунтується машинне навчання, доволі великий, та знаходження рішення задач класифікації продовжує залишатися нетривіальною задачею[25].

З основних проблем можна виділити: складні перетворення даних в кількісні шкали, вилучення значущих факторів в даних та попередня класифікація даних на класи. При цьому велика кількість дослідів показує, що створення універсальної системи розпізнання зображень, без конкретизації під задачу є невирішувальна. Основним підходом вирішення проблеми є створення систем під кожну конкретну задачу, коли кожна клінічна задача вирішується окремо, а потім їх можна або забрати в одну групу, або спробувати узагальнити отримані данні для більш широкої групи захворювань.

Автоматизовані методи аналізу на даний момент починають розроблятися та впроваджуватися з практичними цілями, більшість дослідів так і залишаються лише дослідями і не отримують клінічного використання. Це ще одна велика проблема в доказах ефективності методів та довіри до автоматизованих систем прийняття рішень. Але при цьому багато науковців пишуть про необхідність розширення сфери використання термограм та автоматизації їх аналізу[26].

На даний час є багато успішних прикладів вирішення подібних задач аналізу термознімків. Якщо узагальнювати алгоритм автоматизації аналізу, то можна скласти наступний алгоритм: знаходження області аналізу, застосування певних статистичних функцій для перетворення цієї області в кількісні данні, це може бути текстурний аналіз, насиченість кожного елемента, частота його знаходження на зображенні та багато іншого, це всі кількісні данні називають признаками, за якими потім складуть основу для побудови математичних моделей чи класифікаторів, головна ідея яких є виконання функції прогнозування чи класифікації даного

об'єкта до певної групи, або конкретно по захворюванням, або в більш загальному випадку «патологія» чи «норма».

На першому етапі під час виявлення області інтересів використовують різні методи, але зазвичай це знаходження замкнутих контурів та алгоритми кластеризації. Якраз за допомогою кластеризації виявляються області гіпотермії, та може оцінювати її форма, розміри та інші критеріїв. Або додатково з термознімками можливо використання класичних оптичних знімків тієї ж самої області, що допомагає точніше отримувати області, які треба дослідити, завдяки створенню маски з оптичного знімка і її переносом на термознімок[27].

Під час наступного етапу – збору ознак, можуть використовуватися різноманітні методи: вейвлет-аналіз, текстурний аналіз, курвлет-перетворення. Підсумковий набір даних представляє сукупність кількісних вимірювань всіх наявних зображень. Якщо компонент вийшло занадто багато, то використовують метод головних компонент, який здатен виділити найбільш важливі ознаки для подальшого аналізу. При цьому є ймовірність загубити корисні данні, які можуть стати важливими для класифікації, але якщо цього не робити виникає інша проблема – велика розмірність ознак. Це не дасть змогу алгоритму класифікації працювати швидко, та отримані вихідні моделі будуть перевантажені змінними. При використанні логістичної регресії, перцептрона, рандомного лісі, дерева рішень чи інше, як класифікатора, буде потребувати великою обчислювальною потужності від комп'ютерів та довгого часу обробки.

Але цей етап не завжди є необхідним. Так наприклад, при використанні нейромереж його можна пропустити. Нейромережі самі формують набори важливих ознак. Якщо розглядати навчання з вчителем, то на початкових умовах нейромережі необхідно ввести лише набір даних і міткою до якого класу відноситься той чи інший об'єкт та налаштувати необхідні параметри самої мережі, кількість слоїв, ваги, архітектуру та інше[28].

Після навчання нейромережі чи будь-якого іншого класифікатора необхідно визначити достовірність аналізу, для цього цьому йому подається набір даних, який

не брав участі у навчання і які класифікатор бачить вперше, і на основі отриманих результатів робить додатковий аналіз.

Зазвичай для постановки результативності роботи класифікатора та аналізу його компетентності для використання використовують наступні критерії:

- точність – загальна кількість правильних класифікованих випадків з усіх наявних прикладів;
- чутливість (%) – здатність коректно класифікувати випадки наявності патології;
- специфічність (%) – здатність коректно класифікувати випадки відсутності патології.

Таким чином самі ці критерії стануть основними при оцінці точності класифікатора[29].

Висновки до розділу 1

В першому розділі було проаналізовано сфери застосування термограм, визначено актуальність та розповсюдженість даного напрямку, а також розглянуто ряд проблем, які можуть виникати при його застосування. Окрім цього, визначено основні підходи для автоматизації обробки термзнімків, критерії, на які треба звернути уваги при проектуванні та розробці подібних систем та можливі напрями їх розвитку.

РОЗДІЛ 2

МАТЕРІАЛИ ТА МЕТОДИ

2.1 Матеріали

Як було зазначено раніше, перед початком операції серце необхідно охолодити, а після операції його необхідно нагріти, при цьому відсутні можливість кількісно виразити успішність операції. Тому виникає необхідність в порівнянні даних до та після операції, аби на їх основі мати змогу зробити висновок результативності операції. Було прийнято рішення в створенні системи, яка показує ймовірність того, що в кожен момент часу на знімку можливий різкий перепад градієнта температури чи ні та подальше їх порівняння. Для чого потрібно спочатку створити класифікатор, який і буде виконувати основний розподіл зображень по класам.

Під час операції проводилися дистанційні виміри температури серця, за допомогою термографа фірми FLIR, наразі це є світовий лідер в виготовленні інфрачервоних камер, які включають велику точність результатів, чутливість мінімум в 0.1 градус Цельсія та широким функціоналом та задовільним діапазоном розпізнання температури. Таким чином було отримано 6 відео з гіпертермією та гіпотермією окремо для лівого та правого шлуночку.

Перед початком аналізу даних та проектуванні нейронної мережі, необхідно провести препроцесінг вхідних даних. Тобто, за необхідності, нормалізувати їх, привести до однакової розмірності, позбавитися від артефактів і наявних шумів та виділити області інтересів. Зазвичай для цього використовують різні види фільтрів чи ручне редагування, але при цьому треба бути обережним аби через фільтрацію чи згладжування не видалити з даних важливі, інформативні області.

Також необхідно брати до уваги обладнання, на якому робилися виміри, яка в нього точність, частота зйомки, додаткові налаштування та інші параметри, які можуть впливати на точність та якість отриманих термознімків. В деяких випадках, при використанні якісного обладнання, яке відповідає під вимоги поставленої

задачі, то етап додаткової попередньої обробки може бути пропущений, та є можливість одразу переходити на наступний етап безпосереднього аналізу та інших процесів. Вихідні дані, які використовувалися, мають вигляд, який наведено на рисунку 2.1

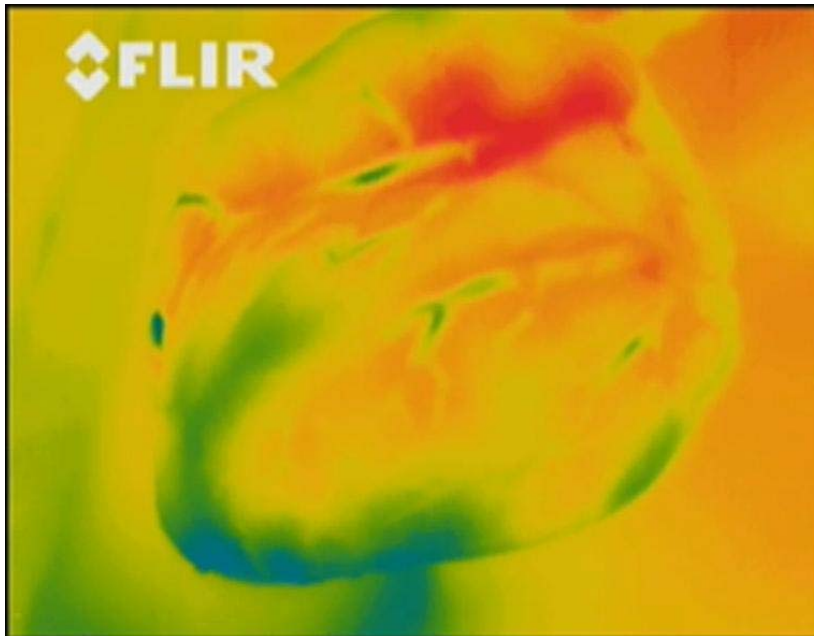


Рисунок 2.1 – Вихідний знімок серця

Через те, що початкові дані були надані в форматі відео .mp4, а для навчання нейромережі необхідно використовувати картинки .jpg, першим етапом є кадрівання відео через кожні 5 секунд та їх розподіл на дві групи «патологія» та «норма», аби мати змогу на цій основі побудувати класифікатор. За значення патології приймається різниця градієнту температури на знімку, яка більше 3 градусів Цельсія, причини вибору такого порогу описувалися в теоретичній частині роботи.

Всі записи робилися на якісні, спеціалізовані апарати, через що, етап попередньої обробка та фільтрації від шумів та артефактів можна пропустити. При роботі з серцем доволі складною є задача позбавлення від артефактів, бо серце може мати свої анатомічні особливості, рубці, які безпосередньо впливають на можливість рівномірного кровопостачання органу, та є важливим класифікаційним маркером.

Основною задачею на цьому етапі є виявлення області інтересів. Дані знімки робилися під час операції на відкритому серці, задля того, аби нормалізувати коронарний кровоток, тому основною областю стануть саме коронарні артерії, бо саме від їх пропускної здатності буде безпосередньо залежати можливість забезпечення серця необхідною кількістю кисню для його нормальної працездатності.

Таким чином, дані знімки необхідно фрагментувати. Для цього кожне зображення необхідно відформатовати та обрізати під однаковий розмір, аби привести дані до максимально рівноцінних умов для класифікації, бо будь-яка додаткова різниця в знімках може сильно вплинути на кінцевий результат.

Також розмір зображень має бути одночасно не занадто великий, аби не перевантажувати електронно-обчислювальну машину при запуску алгоритму кожного разу, але і бути не занадто малим, аби не обрізати інформативні частини знімку. Було прийнято рішення обрати розмірність 250x380. Це дозволяє повноцінно позбавитися від фону, аби він ніяк не впливав на результати класифікації (рис.2.2).

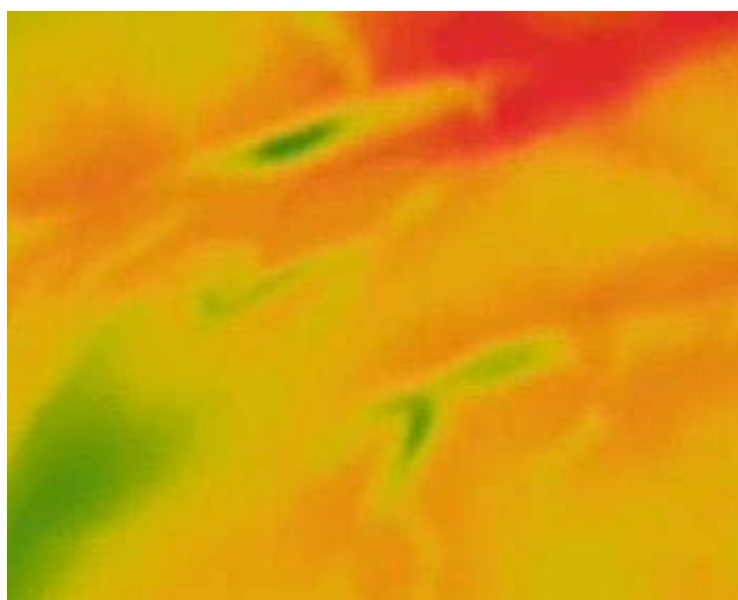


Рисунок 2.2 – Сегментоване зображення термограми серця

Таким чином отримано 105 зображень, які в підсумку було розділено на дві групи «патологія» та «здорові». Також виникла необхідність видалити деякі

зображення, бо вони сильно відрізнялися від інших та могли призвести до викривлення алгоритму класифікації, бо вони були майже повністю однорідні без видимої структури серця чи коронарних судин. Таким чином залишилося 96 зображень.

До першої групи з 64 зображень, відносяться ті термограми, де можна спостерігати градієнт зміни температури вище 3 градусів Цельсія (рис.2.3).

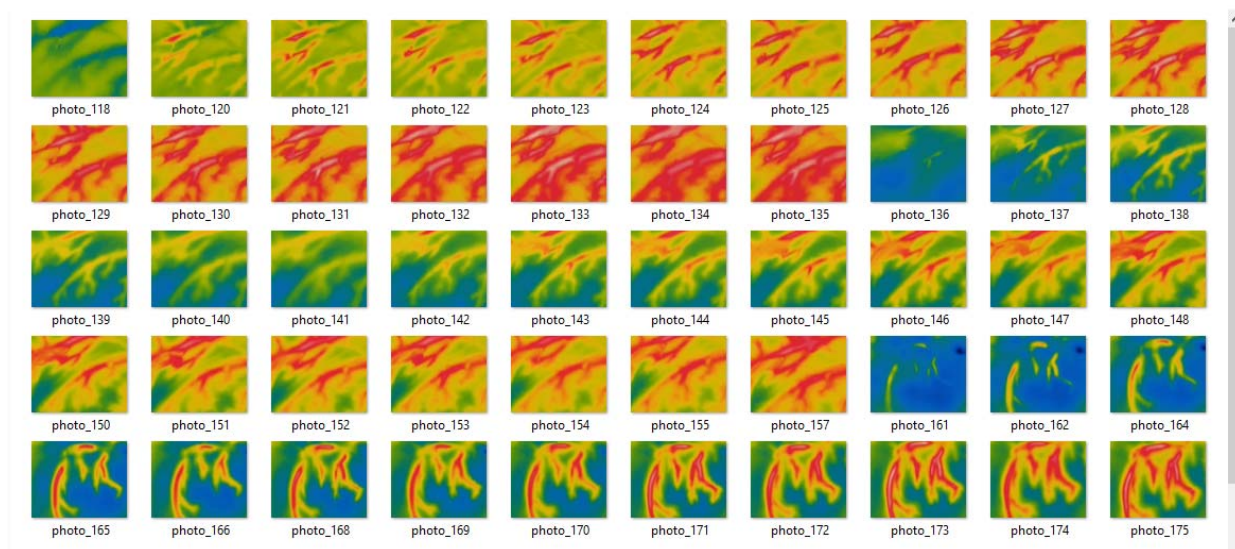


Рисунок 2.3 – Зображення віднесені до класу «патологія»

До другої групи з 24 елементами було віднесено відповідно ті термознімки, де градієнт температури сягав менше 3 градуси Цельсія (рис.2.4.).

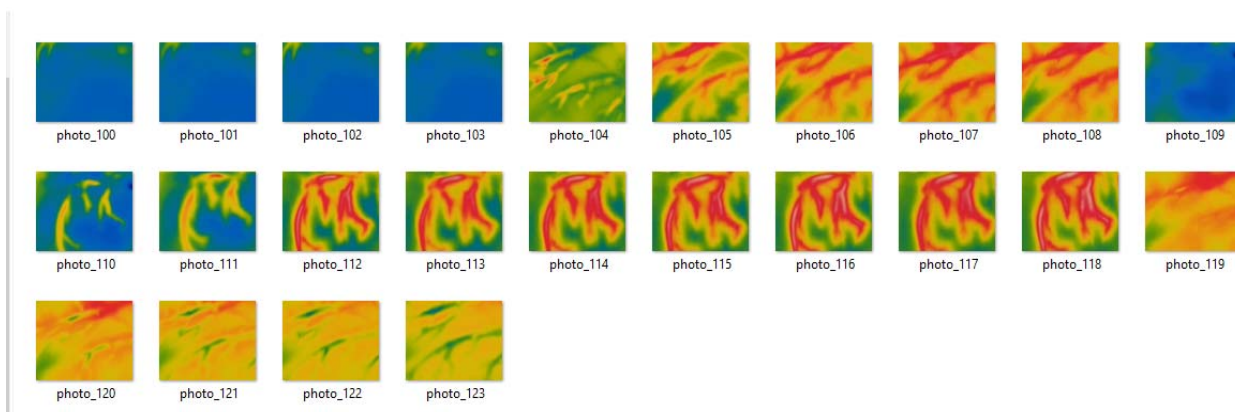


Рисунок 2.4 – Зображення віднесені до класу «патологія»

Останнім етапом попередньої обробки даних є розподіл всередині двох класів на навчальний та валідаційний набір. До першого набору даних віднесемо

70% з кожної виборки (45 та 17 зображень відповідно) на яких буде тренуватися алгоритм класифікації, а інші 30% будуть використовуватися як данні для перевірки достовірності результатів, бо для алгоритму це будуть нові данні, які він ніколи не зустрічав, а для класифікації більш важливіше як відбувається віднесення до категорій нових даних, а не тих на яких відбувалося навчання.

2.2 Методи

Нейронні мережі мають велику популярність, як засіб для аналізу зображень, та медична сфера не стала для цього виключення. В багатьох статтях є опис їх використання для ультра-звукових знімків, термограм, комп'ютерних томограм та інших медичних даних. Вони дають змогу виділити область інтересів та знайти функціонально важливі частини знімка, за якими можна зробити висновок про наявність та відсутність захворювань.

Основним недоліком цього методу є те, що для побудови моделей класифікації витрачається велика кількість обчислювальних ресурсів та для адекватних висновків необхідно на всіх, під час навчання, подати великий масив даних. Але незважаючи на це, нейронні мережі лише продовжують набирати популярність, бо здатні забезпечити високу точність класифікації[30].

Нейронна мережа складається з послідовності нейронів, які з'єднані синапсами за аналогією до біологічних нейронних мереж. Така мережа представляє собою математичну модель, яка має певні налаштування для вирішення конкретної задачі[31].

Нейрони в мережі можна поділити на три типи: вхідний, скритий та вихідний. Коли нейронів є велика кількість, то їх об'єднують в нейронні шари, тобто формується вхідний шар, на який поступає інформація, довільна кількість скритих шарів, де інформація обробляється та трансформується, та вихідний шар, який видає результат роботи нейронної мережі.

Окрім цього, кожен нейрон має два параметри так звані вхідні та вихідні данні. До вхідних даних потрапляє сумарна інформація з усіх попередніх шарів, з

яким з'єднано нейрон, далі ця інформація нормалізується та проходить через функцію активації, де приймається рішення стосовно яке значення буде надаватися до вихідного поля, така функція активації може мати різні форми та залежить лише від вимог до системи.

Також важливу роль в нейронних мережах відіграють синапси, які створюють зв'язок між нейронами. Основним параметром в синапсах є вага, яка вказує на важливість того чи іншого признака та виконує основну роль в обробці інформації[32].

Синапсні зв'язки з додатнім значенням ваг називають збуджуючі, в з від'ємним – гальмівні. Кожен нейрон при обчисленні вихідного значення базується на функції, яка працює зі значеннями ваги та зміщенням, за рахунок ітеративного налаштування яких відбувається навчання нейронної мережі.

Основною особливістю нейронної мережі згортки є те, що вектор ваг та зміщення, які формують фільтр, можуть бути однаковими для різних нейронів, що призводить до скорочення затрати на пам'ять[33].

Таким чином, маємо наступну картину роботи мережі: синапси множать вхідний сигнал на певне число, суматор сумує всі вхідні сигнали до нейрона та ця величина поступає на функцію активації, яка перетворює значення після суматор на іншу вихідну величину в залежності від налаштування функції активації.

В загальному випадку, нейрон можливо повноцінно описати лише за допомогою ваг та функцією активації. За функцію активації було прийнято гіперболічний тангенс, який має значно швидший збіг ніж сигмоїда, та має наступну формулу:

$$\tan(z) = \frac{2}{(1 + e^{-2z})} - 1 \quad (2.1)$$

Ще одною важливою характеристикою для нейронної мережі є значення помилки, тобто наскільки очікуванні результати відрізняються від отриманих. З кожною епохою навчання помилка має мінімізуватися та йти на спад. Є декілька

варіантів підрахунків помилок, які мають ряд переваг та недоліків. Як основний вид розрахунку помилок використовується Mean Squared Error:

$$\text{MSE} = \frac{(i_1 - a_1)^2 + (i_2 - a_2)^2 + \dots + (i_n - a_n)^2}{n} \quad (2.2)$$

Для вирішення задачі використовувалися нейронні мережі з вчителем. Це означає, що при навчанні для всіх вхідних даних існує вихідний вектор значень, до якого класу необхідно віднести кожен вхідний масив інформації. Кожне таке значення послідовно проходить через нейронну мережу та кожен раз ваги та відхилення корегуються, таке продовжується до тих пір поки значення помилки не стане задовільним[34].

Таким чином, основною унікальністю нейронних мереж є те, що вони з самого початку базуються на реальних даних і вже до в коло них будуються моделі для класифікації. Через те, набір даних під час навчання має бути максимально різноманітний та включати до себе дані всіх вихідних класів.

Всі мережі згортки складаються з декількох різних шарів, які можуть чередуватися між собою в довільному порядку. Це шар згортки, субдискретизуючий або агрегуючий шар та повнозв'язний шар, загальний вигляд архітектури представлено на рисунку 2.5. Шар згортки, який є основним для даного типу мереж, являє собою скалярний добуток або взаємну кореляцію та є свого роду фільтром.



Рисунок 2.5 – Архітектура нейронної мережі згортки.

Під час розробки мережі згортки на вхід поступає масив даних, які зберігаються в тензорах – клас у вигляді масиву даних, який має схожу будову з NumPy. Даний тензор буде мати наступну форму: (кількість зображень) \times (висота зображення) \times (ширина зображення) \times (глибина зображення). Далі після проходження скритих слоїв та згортки видозмінюється в карту об'єктів або так званих карти ознак, таким чином після кожного такого проходу, розмірність зображення зменшується та на ньому залишаються тільки важливі для класифікації частини. Надалі вся інформація зберігається в формі матриці наступної розмірності (кількість зображень) \times (висота карт об'єктів) \times (ширина карт об'єктів) \times (канали карти об'єктів).

Шар згортки можна представити наступним чином (рис.2.6).

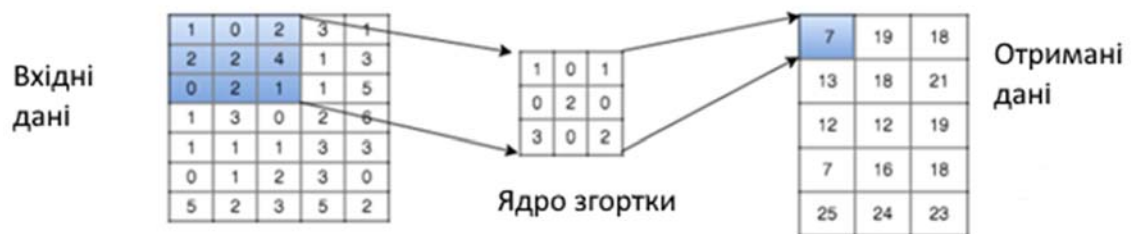


Рисунок 2.6 – Шар згортки

Шар згортки змінює вхідні данні зображення завдяки ядру, яке представляє свого роду налаштовуваний фільтр, в залежності які дані розцінюються як більш важливі, буде відбуватися перетворення матриць. Однак лишається окрім цього необхідність у великій кількості нейронів, через те що зображення має велику кількість вхідних пікселей, які потенційно можуть бути інформативно важливими для класифікації.

Задля зменшення розмірності зображення існує агрегаційний шар або субдискретний. На попередньому кроці, під час згортки виявляються певні важливі елементи на зображенні та посилюються, отже з'являється можливість зменшити розмірність вихідної матриці, аби на наступному етапі прискорити обробку інформації і також така фільтрація допомагає усунути зайві елементи та уникати перенаванчання.

Зазвичай для субдискретного шару використовуються малі вікна, стандартним є матриця 2×2 , які не накладаються одне на одне, завдяки такому підходу зображення зменшується в двічі, основується на виборі максимального значення з тих, які потрапляють у вікно.

Схематичний приклад шару агрегації наведено на рисунку 2.7.

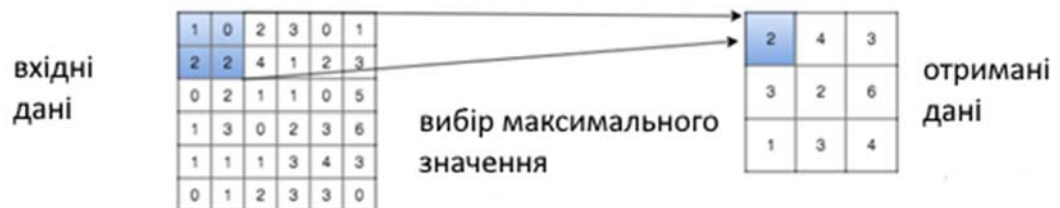


Рисунок 2.7 – Субдискретний шар

Останнім шаром є повнозв'язний шар, який виконує функцію класифікатора та зазвичай має вигляд стандартного перцептрону. Вихідний шар має бути пов'язаний з усіма попередніми шарами, а кількість його нейронів дорівнює кількості шуканих класів. Але при випадку коли є лише два можливих варіанти «норма» та «патологія», можна скоротити кількість нейронів до одного завдяки використанню функції активації гіперболічний тангенс, який дозволяє отримувати дані -1 або $+1$. Таким чином, коли на виході повертається значення -1 , то це означає, що дане зображення відноситься до «патології», а значення $+1$ – «норма»[35].

Висновки до розділу 2

В даному розділі були описані дані, які використовуються для побудови класифікатора, писані які маніпуляції з ними було проведено. Також описана логічна структура основного методу класифікації – нейронної мережі згортки, зазначені всі її основні елементи, структурні блоки, на які буде звертатися вага при проектуванні мережі.

РОЗДІЛ 3

ПРАКТИЧНА ЧАСТИНА

3.1 Розробка нейронної мережі для аналізу термограм

Швидкість роботи нейронної мережі на пряму залежить від кількості оброблених знімків та від їх розміру. Вхідні дані представлені серією зображень розміром 250x380, якщо таку розмірність залишати в продовж всієї роботи мережі, то це призведе до підвищення швидкості обробки та неможливості, але при цьому якщо розмір зображення обрати занадто малий, то на знімку не опиняться важливих частин, для діагностики. Окрім цього, кожне зображення має ще додаткові параметри, такі як канали кольорів, в цьому випадку це червоний, синій та зелений. Тобто фактично одне зображення перетворюється на три при подальшій обробці[35].

Всі зображення потрапляють на вхідний шар, який складається з кількості карт, відповідній кількості вхідних каналі, в цьому випадку їх три для червоного, синього та зеленого кольору.

Після цього всі вхідні дані треба нормалізувати та привести в значення від 0 до 1, використовуючи наступну формулу:

$$f = \frac{p - \min}{\max - \min} \quad (3.1)$$

Де, p – поточне значення пікселя,

\max – найбільше значення, яке може приймати піксель,

\min – найменше значення, яке може приймати піксель.

Наступним етапом роботи є проходження зображень через шар згортки, який має певне ядро. Це ядро представляє з себе фільтр, який буде певним чином змінювати зображення за допомогою перетворень.

Кількість таких шарів визначається для кожної задачі індивідуально та залежить від поставлених задач в швидкості обробки та продуктивності, для розрахунку їх розмірів використовують формулу:

$$(w, h) = (mW - kW + 1, \quad mH - kH + 1) \quad (3.2)$$

Де (w, h) – розмір карти згортки,

kW – ширина ядра,

mW – ширина попередньої карти,

kH – висота ядра,

mH – ширина карти.

За замовчанням всі початкові значення дорівнюють 0. Після цього значення ваг генерується довільним чином від -0.5 до 0.5 та ядро проходиться по всьому зображенню виконуючи операцію згортки наступним чином:

$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l] * g[k, l] \quad (3.3)$$

Де, f – вхідне зображення,

g – ядро згортки.

Шар агрегації теж можна поділити на карти, кількість яких збігається з кількістю шарів на попередньому етапі. Як було зазначено раніше, головна задача цього шару зменшити розмірність зображення через те, що всі важливі елементи були виділені на попередньому етапі, зараз можна позбутися від зайвих елементів, які можуть спричинити перенаванчання. Таким чином, на виході отримується нове зображення з виділеними основними елементами та повторюючи попередні кроки вони знов аналізуються, знаходяться нові важливі елементи та на останніх етапах вони формуються в характеристичний вектор для класифікації[36].

Для цього також використовують ядро певної форми, але основна його особливість в тому, що відбувається не зміна попереднього зображення, а

трансформація з вибором максимального елемента з всі наявних, які потрапили до ядра. Такі фільтри стандартно мають розмір 2×2 , що призводить до зменшення зображення в двічі. В окремих випадках це ядро може встановлюватися розміром 3×3 та це залежить лише від поставленої задачі та вхідних даних. Загальним виглядом, формула для розрахунку має наступний вигляд:

$$x^l = f(a^l * \text{subsample}x^{l-1} + b^l) \quad (3.4)$$

Де, x^l – вихід шара l ,

$f()$ – функція активації,

a^l, b^l – коефіцієнт зсуву l ,

$\text{subsample}()$ – операція вибору локального мінімуму.

Нейрони агрегаційного шару пов'язані з нейронами скритого шару та дорівнюють кількості елементів один одного, але при цьому з'єднання нейронів може відбуватися довільним чином. Обчислювальні значення рахуються за формулою:

$$x_j^l = f\left(\sum_i x_i^{l-1} * w_{i,j}^{l-1} + b_j^{l-1}\right) \quad (3.5)$$

Де x_j^l – карта ознак,

$f()$ – функція активації,

b^l – коефіцієнт зсуву,

$w_{i,j}^l$ – матриця вагів.

Останнім є вихідний шар, де кількість нейронів дорівнює кількості шуканих класів, але при випадках, коли класи лише два, то може використовувати один нейрон. В цьому випадку класифікація буде проходити наступним чином, що за функцію активації необхідно взяти гіперболічний тангенс, який може приймати як від'ємні так і додатні значення, отже на виході при -1 буде відбуватися присвоєння

зображення до першого класу, а при виході +1 зображення приписується до другого класу.

3.2 Тестування та аналіз класифікаторів

Таким чином після побудови всіх класифікаторів необхідно пройти етап навчання. Для цього на моменті підготовки розподілу термознімків їх було поділено на два класи «норму» та «патологію». Для того, аби проаналізувати наскільки якісно нейронні мережі будуть працювати під час роботи з новими знімками, необхідно проводити навчання та тестування нарізних вибірок. Для цього ділимо кожен з класів на дві підгрупи у співвідношенні 70/30.

Отримаємо, що під час навчання необхідно слідкувати за значення функції помилок, яка має досягти мінімального значення, або вийти на плато та більше не знижати свого значення, що буде свідчити про завершення етапу навчання. І після цього перевірити за допомогою даних, які ще не використовувалися. Таким чином для користувача важливіше, саме точність розпізнавання нових зображень, а не загальне значення функції помилок для мережі. Адже ці значення можуть не співпадати, при цьому загальні характеристики нейронної мережі даватимуть високий результат точності тільки через те, що вони перенавчалися та виявилися сильно специфіковані для поточних даних і не здатні працювати з новими, що є неприпустимо для даних систем[37].

Після того як класифікатори було розроблені та пройшли етап навчання, необхідно повірити їх точність та обрати основний класифікатор, який буде використовувати в системі.

Для того аби перевірити результативність класифікатора використаємо наступні критерії: точність, чутливість та специфічність. Та також основні показники загального відсоток правильної класифікації (TP – true positive, TN – true negative) та наявність помилок першого та другого роду (FP – false positive, FN – false negative).

На рисунку 3.1 представлена робота трьох видів нейронних мереж: дві мережі згортки на 5 та 8 шарів та нейро-нечітка мережа CFNN.

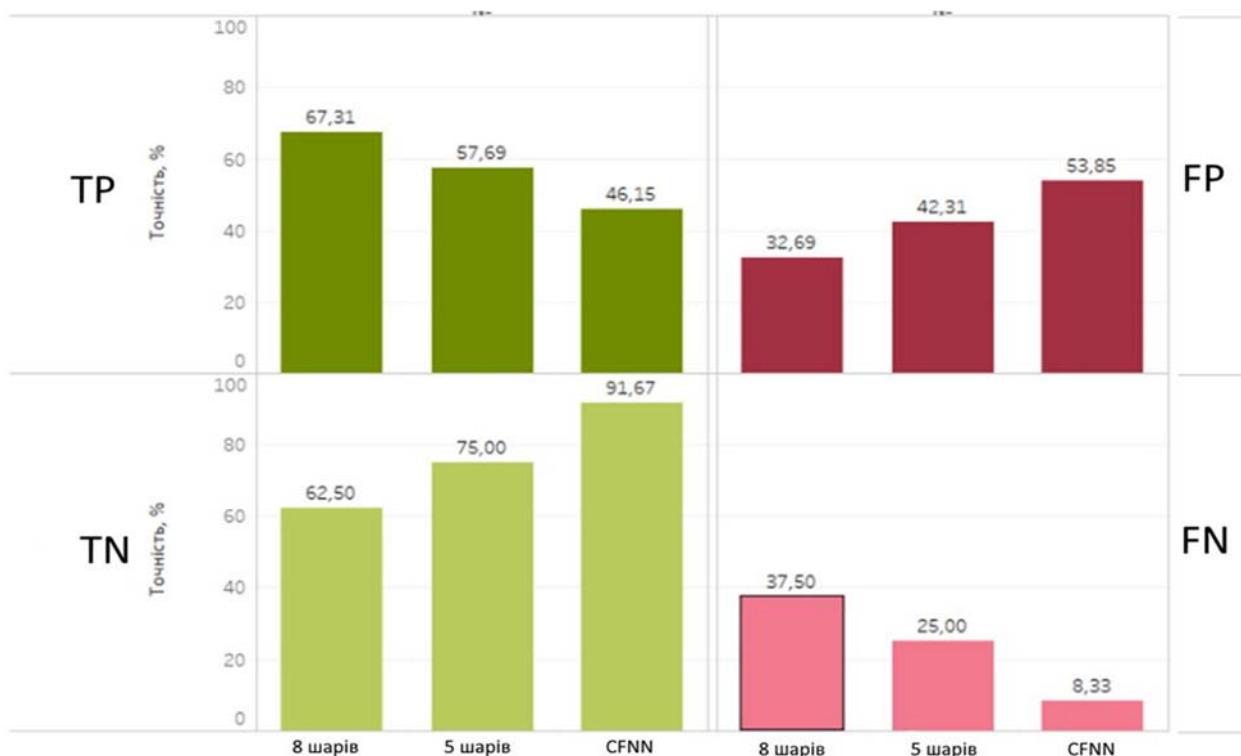


Рисунок 3.1 – Аналіз роботи нейронних мереж.

Виходячи з рисунка бачимо, що нейрона мережа на основі 8 шарів дала найкращий результат до розпізнання наявності патологій (TP = 67,31%) та майже такий саме результат для виявлення норми (TN = 62,5%), при чому дві інші мережі мають протилежний результат, вони гірші для виявлення патології, але гарно розпізнають норму і досягають точності 91%, але лише гарного визначення норми недостатньо як основного критерія вибору класифікатора. На даному етапу важливо зберігати баланс між точністю розпізнавання кожного з класів.

Але при цьому, необхідно звертати увагу на можливість помилок першого та другого роду. В даному випадку розглядання нейро-нечіткої мережі не є можливим для використання, адже вона допускає багато помилок, коли при наявній патології відносить зображення до класу норми (FP = 53,85%). Подібний відсоток помилок є недопустимим для медичної сфери та дає багато результатів не до діагностики, які можуть стати фатальними для пацієнта. В цій сфері краще мати гіпердіагностику, коли норму відносять до патології, адже існують інші додаткові засоби діагностики

для уточнення результату, але при не до діагностиці, будь-яке втручання припиняється та вся інформація про можливі патології залишається не виявленою.

Так як, дана класифікація призначена як допоміжна система для лікаря, під час відкритих операції на серці, особливу увагу необхідно звернути саме точності розпізнання патологій та ймовірності віднести патологію до норми, таким чином найкращим варіантом є використання мережі згортки з 8 шарів.

Окрім цього, розглянемо також інші статистичні критерії порівняння класифікаторів, наведених в таблиці 3.1.

Таблиця 3.1 – Порівняння якості нейронних мереж

№	Чутливість	Специфічність	Точність
5 шарів	57,69	75,00	63,15
8 шарів	67,31	62,50	65,78
CFNN	46,15	91,67	60,52

В даній таблиці наведено загальні значення розпізнання норми та патології та також підрахована загальна точність, на основі якої видно, що всі класифікатори мають дуже подібну точність, але при цьому суттєво різні відсотки розпізнавання в середині кожного з класів.

Отже, враховуючі всі дані, можна вважати, що оптимальним варіантом використання є нейрона мережа згортки з 8 шарів.

3.3 Нечітка математика

Окрім цього, для медицини результат точності в 65% не є повністю задовільним та вимагає розгляду методів покращення, які будуть наведені далі в розділі, та для їх проектування необхідно використати певні математичні розрахунки на основі нечіткої логіки, яка буде слугувати основою для визначення ймовірності наявності патології на знімках[38]. Через те, що прямі математичні розрахунки дають гарну точність класифікації, але їх використання для обробки

великих масивів даних вимагає великих витрат потужності, то їх використання є доцільним лише як окремого модуля, для ведення незначних розрахунків.

Функція належності ставить у відповідність деякому числу з інтервалу від 0 до 1 його належність до іншої множини у вигляді:

$$\mu_A(x) \in [0, 1] \quad (3.6)$$

Але при цьому треба мати на увазі, що міра відповідності є суб'єктивним показником та визначається окремо і результаті оцінки експерта, та вона має інше значення ніж ймовірність події, адже друга будується на стохастичній невизначеності.

При побудові подібних функції відбувається асоціація властивостей певної множини, яка характеризує об'єкти. Чим більше певний об'єкт відповідає цим властивостям, тим більша його степінь відповідності та значення ближче до 1 та навпаки.

Тобто крайні значення, коли $\mu_A(x)=1$ означає, що елемент x характеризується цими властивостями, $\mu_A(x)=0$ означає протилежне, що об'єкт x неможливо характеризувати через дану множину.

Функція відповідності може мати різні форми, які підбираються окремо під задачу. Для поточного випадку з градієнтами температури доцільним є використання S-подібної функції, значення якої нагадує сигмоїду та описується наступною формулою:

$$\mu_A(x, a, b) = \begin{cases} 0, & x < a \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-a}{b-a}\pi\right), & a < x < b \\ 1, & x > b \end{cases} \quad (3.7)$$

Де, x – поточне значення змінної,

a, b – довільні числа, при чому $a < b$.

Для простоти розрахунків та зменшення навантаження на систему, подібний вираз можна розбити на підсистему з чотирма параметрами, де розбиття відбувається на рівні відповідності в 0.5, що дозволяє конкретизувати форми кривих для двох окремих частин графіка, коли для випадків, якщо функцію не можливо описати однозначно, буде редагуватися кожна з частин. Таким чином, аналогічно представимо попередню формулу наступним чином:

$$\begin{cases} 0, & x < a \\ 2\left(\frac{x-a}{b-a}\right)^2, & a < x < \frac{a+b}{2} \\ 1 - 2\left(\frac{b-x}{b-a}\right)^2, & \frac{a+b}{2} < x < b \\ 1, & b < x \end{cases} \quad (3.8)$$

В загальних випадках S-функція використовується для представлення невизначеності характеру: високий рівень, більша кількість, вище норми, висока якість. Та її графічне представлення будується у вигляді графіку (рис.3.2).

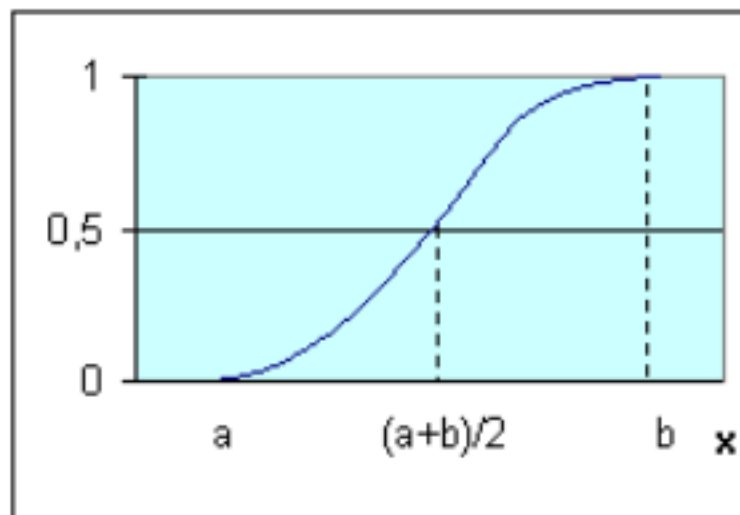


Рисунок 3.2 – Загальний вигляд S-функції

Особливість таких функції є визначення високої степені прояву певної ознаки, яка визначається та функція належності в цих випадках завжди є зростаюча та неперервна.

3.4 Аналіз гістограм

Після того, як було обрано класифікатор, необхідно розробити метод по якому будуть аналізуватися отримані дані. Через те, що візуальний аналіз є не завжди точним методом аналізу, існують вимоги для розробки додаткових критеріїв оцінювання діаграм.

Побудований класифікатор, дає змогу обробляти зображення в динаміці, якщо завантаження зображень робити в строгому порядку, що формує з них діаграми розподілу зон з підвищеним градієнтом та зон без патології. Таким чином можна порівнювати результати, які були до операцій та після, та по вигляду отриманих результатів гістограм та статистичному аналізу робити висновок про результативність операції.

Розглянемо два набора даних, від роботи нейронної мережі до та після операції та проаналізуємо їх за допомогою описової статистики та потім проведемо аналогічний аналіз для даних після роботи з математичною функцією нечіткої логіки.

По-перше необхідно розрахувати основні статистичні параметри, виберемо основні з них: середнє значення, стандартне відхилення та розмах вибірки, на основі яких проаналізуємо динамічну зміну градієнта температури.

Таким чином, в задачі критерієм для визначення успішності операції є те, що середнє значення після операції та стандартне відхилення будуть менші після операції ніж до. Тому що після операції кровопостачання серця нормалізується і ситуації, коли спостерігається висока зміна градієнта температури стає менше, отже саме ці критерії головними для аналітики.

Ці додаткові статистичні показники дають змогу побачити статистично важливі зміни, а при переводі даних в цифри їх стає простіше проаналізувати та порівняти, що стає необхідним, якщо розподіл значень буде занадто неоднозначним та робити висновки по візуальній складовій стає неможливим та не інформативним критерієм.

Порівняльна характеристика для обробки даних динаміці мережею виглядає наступним чином (рис. 3.3).

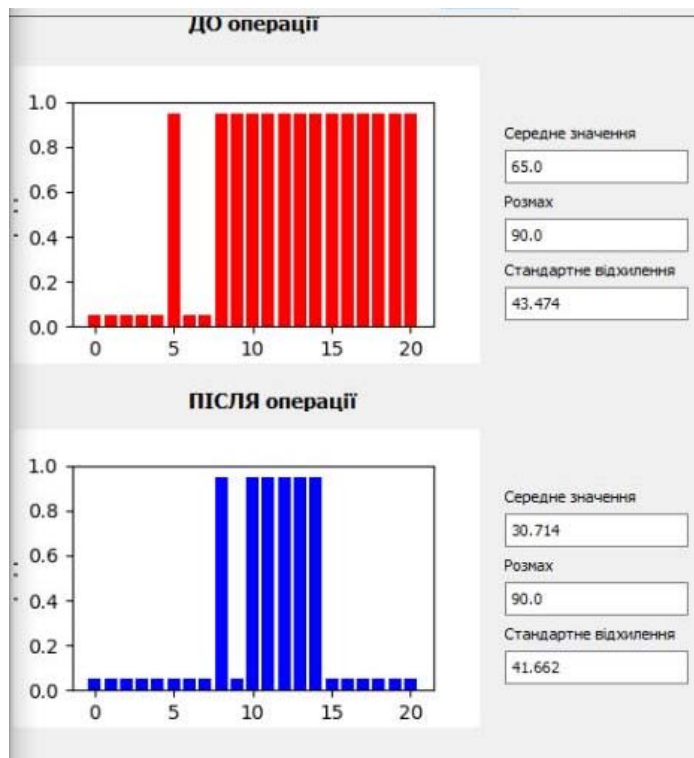


Рисунок 3.3 – Аналіз даних до та після операції нейронної мережі

Висновки, які можна зробити з отриманих даних є те, що і візуально і статистично бачимо, що значення піків суттєво зменшилося після операції ніж було до. Також про це свідчить і статистичні показники, а точніше, середнє значення зменшилося з 65 до 30. Отже даний метод аналізу є інформативним та придатний для подальшої роботи та впровадження в систему.

При цьому розглядаючи значення, які отримуються від нечіткої математики подібних розрахунків стає недостатньо, адже зникає можливість якісно оцінити наскільки одні дані візуально відрізняються від інших, тому для визначення розподілу діаграм та визначення їх однаковості визначимо та застосуємо необхідний статистичний критерій.

Першим етапом необхідно визначити тип даних з якими відбувається робота, кількісні чи якісні вони. В цьому випадку це кількісні неперервні дані. Які за

розподілом не проходять тест на нормальність, тому вибір буде відбуватися з непараметричних критеріїв.

Наступним етапом визначаємо скільки груп порівнюється. В даній задачі існує тільки два набори даних, отже і порівнюватися будуть дві групи. Та визначаємо взаємозв'язки між ними. Через те, що дані отримуються в процесі двох різних процесів, нагрівання та охолодження серця, та неможливо зіставити часові проміжки між собою, то вважається що дані вибірки є незалежними.

Тобто враховуючі всі вхідні дані, необхідним критерієм для порівняння вибірок стає критерій Манна-Уїтні, який є аналогом для критерія Стюдента тільки для непараметричних даних.

Цей критерій ранжирувані та знаходяться в порядковій шкалі, тому першим етапом буде необхідно перетворити неперервні дані в порядкові. Також необхідно звернути уваги на кількість даних в вибірці. Якщо вибірка має більше 20 даних, то розподіл швидко сходиться до нормального розподілу та на цей випадок критерій Манна-Уїтні має окрім p -значення значимості, ще і z -значення, яке використовується в подібних випадках[39].

Висновки по результату тесту робляться аналогічні як і для критерія Стюдента з незалежними вибірками, але при використанні Манні-Уїтні відбувається сумування індикаторів попарного порівняння між вибірками та за своєю характеристикою він є більш ніж критерій Стюдента для аналогічних випадків. Також цей критерій актуально використовувати для невеликих вибірок, розмір яких не перевищує 60 одиниць.

Метод будується на порівнянні перехресних значень в двох вибірках, якщо вони мають велику область перехрещення, то вважається, що вибірки не мають відмінностей. Після цього також відбувається порівняння з емпіричними таблицями для U критерія, який визначає наскільки велика зона однаковості між вибірками.

Таким чином алгоритм розрахунку критерія Манні-Уїтні можна представити наступним чином:

- Об'єднуємо вибірки, при цьому роблячи замітки, які дані відносяться до якої вибірки;
- Проводимо ранжування починаючи з найменшого значення, присвоюючи йому найнижчий ранг;
- Сумуємо для кожної вибірки значення рангів та визначаємо, яка з двох вибірок більша;
- Проводимо розрахунок значення U за формулою:

$$U = n_1 * n_2 + n_x \frac{n_x + 1}{2} - T_x \quad (3.9)$$

Де, n_1 – розмірність першої вибірки,

n_2 – розмірність другої вибірки,

T_x – рангова сума більшого значення,

n_x – більше число з n_1 та n_2 .

- Визначаємо критичне значення U , користуючись таблицею. Чим менше значення U тим достовірність різності вибірок вища.

Таким чином проаналізуємо ті ж самі дані, які були використані для нейронної мережі, але тепер для нечіткої математики та проведемо додаткові розрахунки через статистичний критерій Манні-Уїтні, аби переконатися в тому, що дві вибірки дійсно різняться між собою і ці значення статистично значимі при цьому збережемо розрахунок описової статистики.

Отже о результаті подібних розрахунків було виявлено, що дві вибірки є статистично різними та p -критерій менше за 0.05, при цьому візуальна оцінка в цьому випадку викликає труднощі, адже ранжування можливих значень збільшилося та вибірки не є однорідними. Також дивлячись на середні значення до та після операції бачимо суттєві зміни, коли середнє значення впало з 44 одиниць до 19, і при цьому стандартне відхилення теж зазнало великих змін від 15 до 6.

Таким чином можна вважати що і в цьому випадку розрахунки показали, що операція на відкритому серці є результативною та кровообіг зазнав змін і розподіл температур став більш однорідним. Дані приведені на рисунку 3.4.

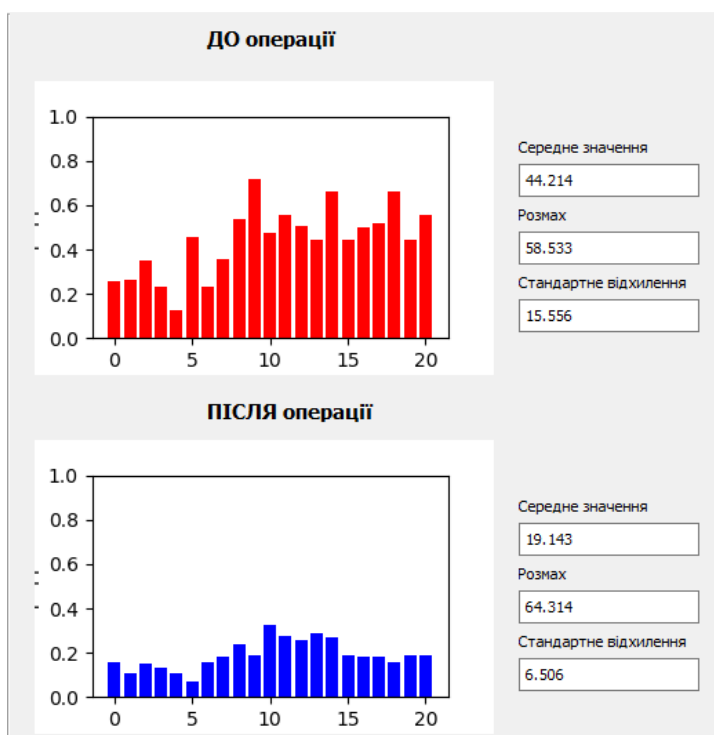


Рисунок 3.4 – Аналіз даних до та після операції нечіткою математикою

Таким чином, під час запуску нечіткої математики, можна досягти точних результатів класифікації та зберігати це термограми як матеріал для подальшого навчання мережі.

3.5 Методи покращення алгоритму класифікації

Як було зазначено вище, то точність, яка отримана за допомогою класифікатора на основі нейронної мережі з 8 шарів для медичної сфери є не повністю задовільна, тому необхідно розглянути додаткові можливості покращення алгоритму та зазначити ці переваги над іншими методами.

По-перше, основна проблем в роботі класифікатора полягає в тому, що для аналізу використовується недостатня кількість зображень, але при спробах штучно їх помножити, за допомогою методів машинної інженерії, то виникає проблема

перенавчання, коли моделі занадто заточені під фізіологію конкретних пацієнтів. Та при цьому наразі немає природніх шляхів для збільшення даних. Адже вона мають бути не тільки від різних пацієнтів, а і обрана вибірка має бути репрезентативною до генеральної сукупності, таким чином, включаючи в себе не тільки гендерну різноманітність, а враховуючи анамнез людей.

Тому, незважаючи на те, що зараз є багато методів розробки нейронних мінімальну кількість навчальної вибірки, доходючи до значень по два зразки для кожної групи, але таким чином, через можливу анатомічну різницю, такі мережі можуть допускатися помилок саме для медичної сфери.

Також, якщо розглядати лише аналіз термограм в медицині, то найпопулярнішими є три методи: текстурний аналіз, математичні розрахунки та нейронні мережі. Через те, що для навчання всіх нейронних мереж необхідна значна кількість матеріалів, то приводити їх в альтернатив методу не є доречним. Необхідно звернути увагу на два інші варіанти.

Дуже гарні результати класифікації, які в багатьох роботах перевищують 90% точності розпізнавання, стосується текстурного аналізу[20]. Це метод аналізу, який за допомогою математичних розрахунків схожості пікселів на обраному відрізьку, розраховує певну кількість характеристик, які потім аналізує класифікатор. Але такий метод більше підходить до однорідних частин тіла, такі як молочні чи щитовидні залози. Для визначення раку в цих випадках основним методом є використання комбінації контурного аналізу, для правильного визначення розміщення органу, та безпосередньо текстурного аналізу, які в сумі дають велику точність та активно використовуються в медичній сфері[31].

Але для використання подібного методу виникає основна проблема – структура серця. В кожній людини можуть бути свої анатомічні особливості, рубці, які будуть сильно впливати на результат аналізу та можуть призвести до багатьох випадків некоректності результатів класифікації, адже однаковість зображення є важливим фактором для цього методу аналізу. Таку проблему можна спробувати вирішити за допомогою розбиття термограми на невеликі фрагменти та аналізувати їх окремо один від одного, що забезпечить наявність певної однорідності тканини.

Та навіть при таких умовах, на зображеннях, де є стики міокарду та коронарних судин буде наявний різкий перехід температури, що алгоритм може розрізнити як патологію. В даному випадку буде важко прослідкувати за логікою виконання аналізу та на якому етапі класифікатор робить висновки.

Останнім методом аналізу залишається введення додаткових математичних розрахунків, які допоможуть підтримати точність роботи класифікатора і при цьому, на відміну від їх самостійного використання, здатні забезпечити швидкість виконання програми.

Основний сенс алгоритму на основі нечіткої математики, принцип та результати роботи якої були описанні вище, є можливість забезпечити нейронну мережу новими розміченими даними, які можна використовувати для до навчання мережі. Таким чином, поки нейронна мережа має низьку точність, її можна використовувати для знаходження швидкого розвитку задач, а при наявності часу математичні функції будуть видавати точний результат. Окрім цього, після кожної роботи нечіткої математики отримані дані будуть зберігатися в окремі файли та розбиватимуться по класам, тоді при оновленні системи, її перезапуску після робочого дня, нейронна мережа, опираючись на нові дані, буде перераховувати власні моделі та самонавчатися, чим підвищуватиме точність класифікатора. Також такий підхід дозволить, в майбутньому, збільшити кількість класів та зробити їх градацію.

Висновки до розділу 3

В даному розділі було реалізовано та порівняно декілька варіантів класифікаторів для аналізу термознімків, серед яких було обрано нейронну мережу на основі згортки з 8 шарів для подальшого впровадження її в систему, яка дає загальну точність в 65%. Також були проаналізовані можливі методи покращення алгоритму класифікації.

РОЗДІЛ 4

ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ

Етап проектування є одним з основних етапів під час створення програмного продукту. Це є першим етапом перед тим, як почати будувати дизайн продукту, писати код та тестувати результати. Зараз існує багато методів та підходів для розробки програмного продукту та його проектування. Необхідно розписати детальний план роботи, терміни та очікувані результати не тільки з точки зору розробника, а і споживача[40].

Складання плану допомагає чітко дотримуватися етапів розробки та оцінити в цілому часозатрати на проект. Також, це дає можливість виключити можливість забути про частину функціоналу, яку необхідно розробити, і уникнути розбіжностей у баченні системи між розробником та кінцевим споживачем.

Окрім чіткого плану, необхідно мати можливість швидко його змінити та додати нові можливості до продукту. Таким чином, зараз частіше використовують технологію Agile, яка дозволяє створювати продукт циклічно, починаючи з базової версії і з часом, без великих зусиль, додавати нові елементи чи виключати старі. Всі етапи проектування, кодування, тестування здійснюються для кожного етапу окремо, даючи можливість вносити необхідні правки.

Але навіть при цьому, певний набір основних діаграм необхідно проектувати на початкових етапах. Зазвичай для проектування використовують стандартний набір UML-діаграм, які розділяються на структурні діаграми, діаграми поведінки та взаємодії, кожен з яких, хоча і схожі між собою, але виконують різну базову функцію, на яку звертається основна увага, та також часто, як допоміжні діаграми, використовують IDEF, які призначені для зазначення процесів та структур з якими буде взаємодіяти майбутній програмний продукт, або з яких він буде складатися[41].

Варіативність діаграм може змінюватися від проекту в проект, але з основних можна виділити наступні: діаграма класів, use-case-діаграма та контекстна діаграма. Ці діаграми здатні надати базове уявлення про програму, але

використання лише їх, буде робити етап проектування неповним, бо не враховуються багато інших, необхідних деталей про ієрархію системи, її взаємодію з іншими, про життєвий цикл об'єктів. Ці всі данні необхідні для правильної розробки і написання коду. Вони дозволяють одразу зрозуміти, які структури даних необхідно використовувати, з яких елементів мають складатися функції та як вони можуть працювати в сукупності з іншими компонентами.

Для проектування даного програмного продукту були обрані наступні діаграми: контекстна діаграма IDEF0, діаграма декомпозиції IDEF0, діаграма IDEF3, дерево вузлів, діаграма класів, Use-case діаграма, діаграма послідовності, діаграма станів[42].

4.1 Контекстна діаграма IDEF0

Беручи до уваги зазначені задачі до програмного продукту, було побудовано контекстну діаграму (рис.4.1) IDEF0[43].

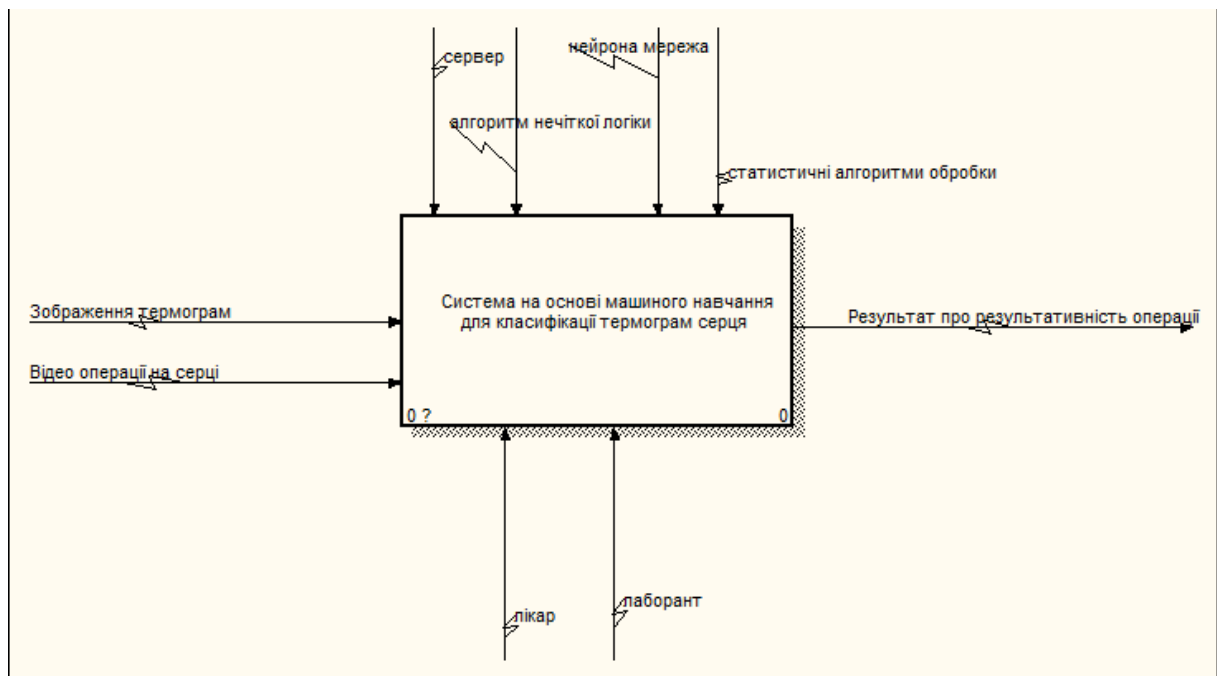


Рисунок 4.1 – Контекстна діаграма IDEF0

На діаграмі можна побачити повну структуру процесів роботи програмного продукту, побачити вхідні, вихідні данні, управління та механізми.

4.2 Діаграма декомпозиції IDEF0

Наступним логічним етапом є побудова декомпозиції (рис.4.2), для попередньої контекстної діаграми, що забезпечить деталізацію процесів роботи продукту. В цьому випадку, загальний процес роботи програмного продукту можна розбити на вісім наступних етапів: окреме завантаження термограм та відео з операції, сегментування знімків, навчання нейронної мережі, збереження отриманих моделей після навчання для подальшого їх використання в класифікації, нечітка класифікація знімків, статистична обробка гістограм які отримали після класифікації відео до та після операції та останнє – виведення висновку стосовно успішності проведеної операції.

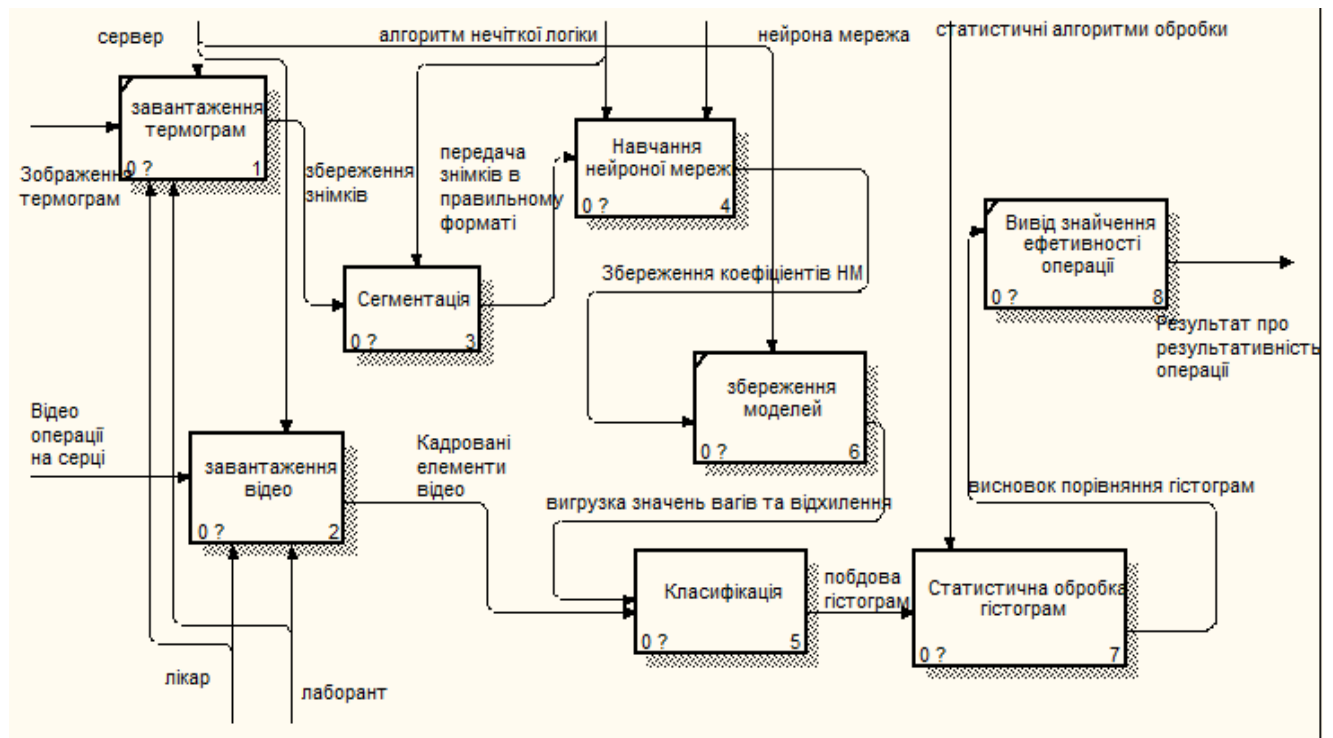


Рисунок 4.2 – Діаграма декомпозиції IDEF0

Також над діаграмою першого рівня теж необхідно провести декомпозицію, аби мати більш конкретне представлення про кожний етап, але замість цього можна використати і сукупність інших діаграм, які в сумі теж дадуть повну картину продукту. Такими діаграмами можуть стати IDEF3, DFD, дерево вузлів та інші діаграми з методології IDEF. Якщо брати найбільшим повний комплект діаграм, то

для кожного рівня декомпозиції необхідно прописувати окремо кожний вид діаграми, який би описував їх бізнес-процеси, з якими файлами працює кожен блок та як вони взаємодіють між собою, в середині яких підсистем знаходяться та інше. Було вирішено надати приклади лише декількох таких діаграм, або зобразити загальну логіку програми.

4.3 Діаграма IDEF3

На діаграмі IDEF3 (рис.4.3) показана деталізація процесу роботи системи з аналітичної точки зору, вказавши послідовність процесів та як вони зв'язані між собою.

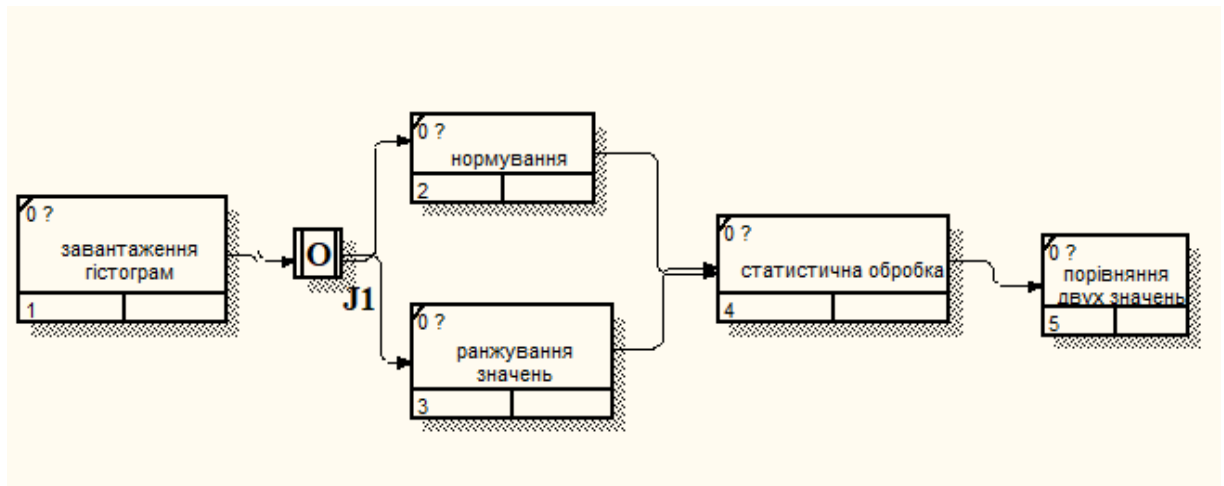


Рисунок 4.3 – IDEF3 для процесу статистичного аналізу гістограм

IDEF3 показує причинно-наслідкові зв'язки між ситуаціями і подіями в зрозумілій формі, використовуючи структурний метод вираження знань про те, як функціонує система, процес або підприємство.

4.4 Дерево вузлів

На діаграмі дерева вузлів відображені всі процеси та під процеси роботи програми. За її допомогою можна оцінити складність виконання роботи в цілому та послідовність виконання процесів (рис.4.4).

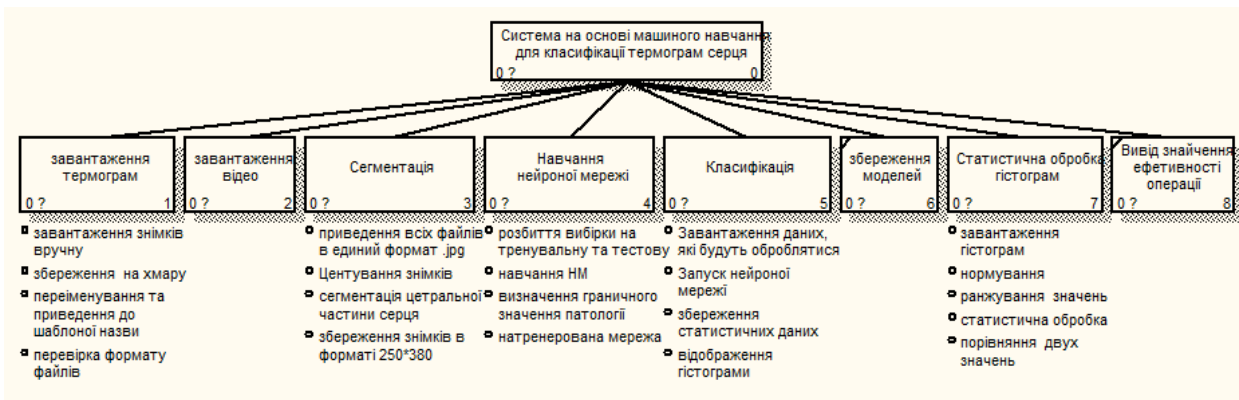


Рисунок 4.4 – Дерево вузлів

На ній можна бачити декомпозиційні частини від діаграми IDEF0, але більш в спрощеному форматі, без додаткових деталей. Також дерево вузлів може мати більшу глибину, під процеси складаються з іншої множини під процесів.

4.5 Діаграма класів

Діаграма класів вважається основною при проектуванні продуктів (рис.4.5).

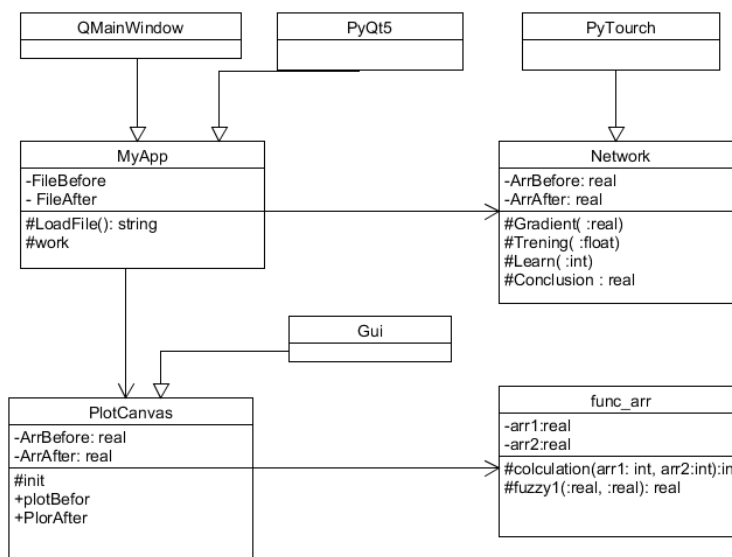


Рисунок 4.5 – Діаграма класів

На ній відображаються основні класи програми, їх функції, глобальні змінні та зв'язок цих всіх елементів один з одним та з іншими стандартними бібліотеками, які використовуються.

4.6 Use-case діаграма

На діаграма Use-case (рис.4.6) відображено взаємодії між актором (користувачем) та інтерфейсом.

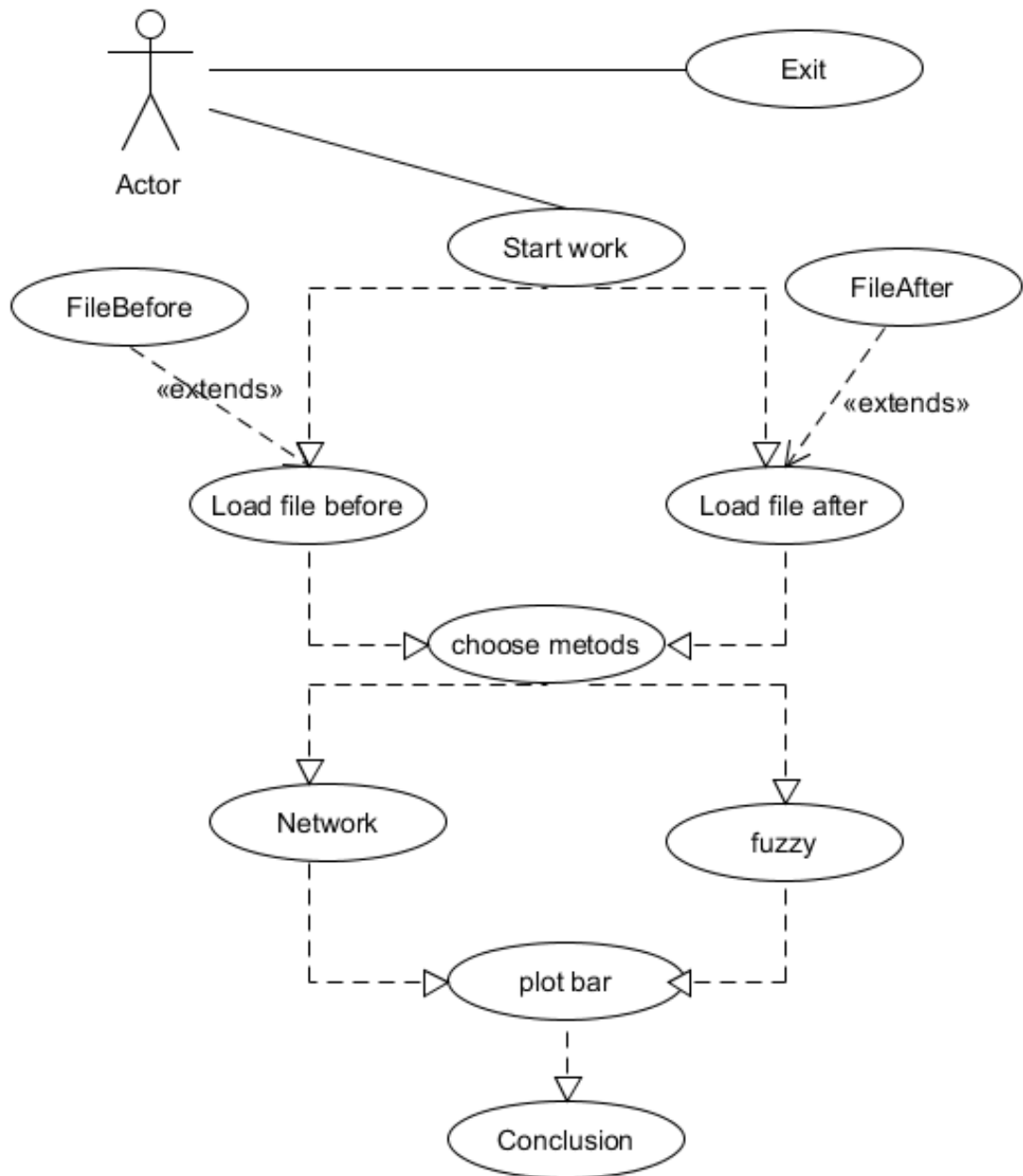


Рисунок 4.6 – Use-case діаграма

До цих етапів було включено наступні елементи: розпочати роботу з програмою або вийти з неї, завантажити відео чи фото, обрати метод для аналізу, ознайомитися з гістограмами та ознайомитися з висновками роботи.

4.7 Діаграма послідовності

Діаграма послідовностей слугує для відображення об'єктів у часі, які повідомлення вони відправляють та в якій послідовності виконуються ті чи інші функції програми (рис.4.7).

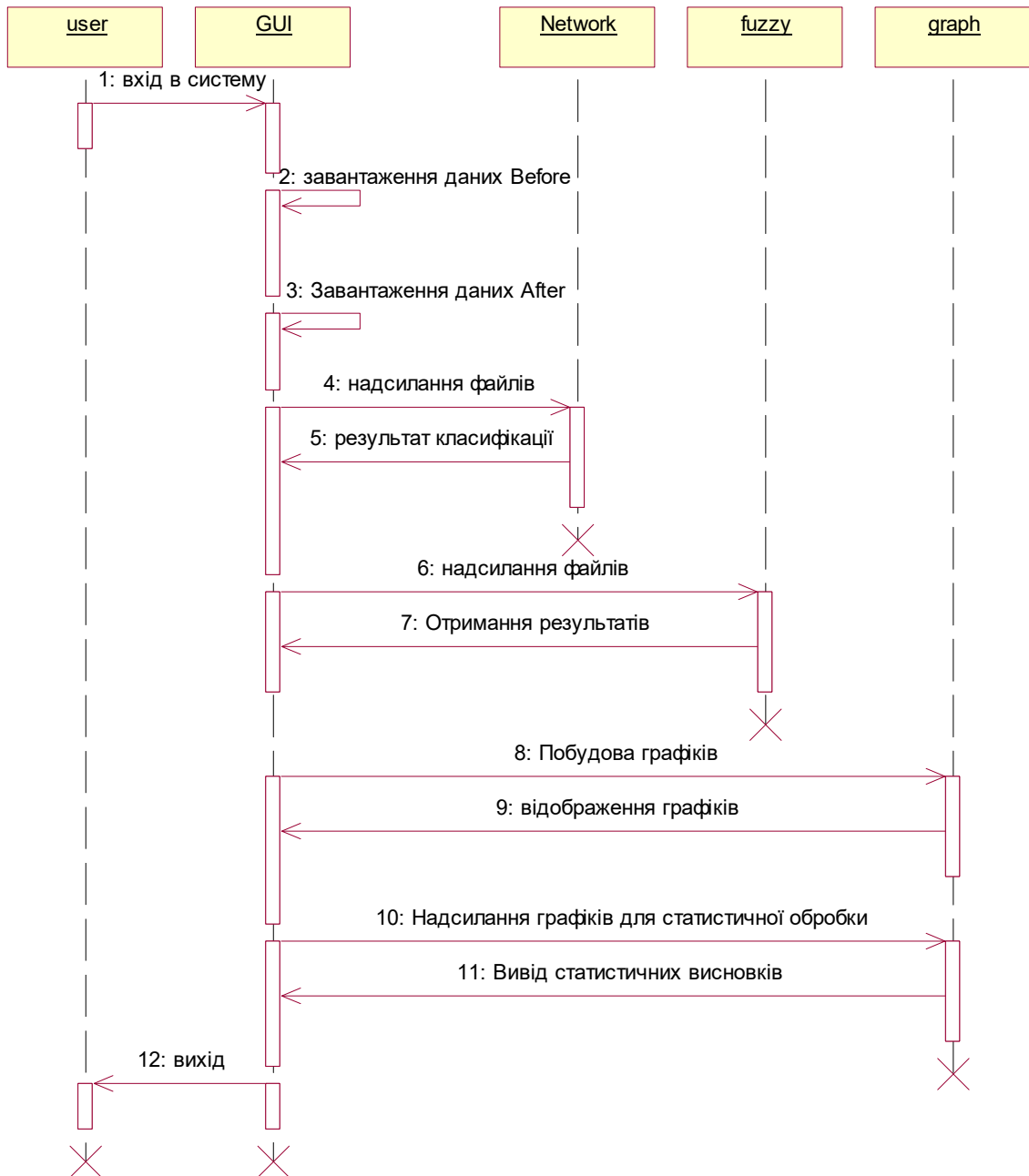


Рисунок 4.7 – Діаграма послідовностей

Таким чином можна відслідкувати як поведуться об'єкти в середині програми та їх взаємодії.

4.8 Діаграма кооперацій

Діаграма кооперацій відображає множену об'єктів та їх взаємодії в загальному вигляді без деталей. На цій діаграмі використовуються дві основні структури: блоки даних та стрілочки чи стрілочки. Блоком показується об'єкт, а лініями позначається з якими іншими об'єктами він може обмінюватися повідомленнями. Також, якщо в програмі передбачено декілька варіантів використання, то для кожного з них будується своя діаграма.

Діаграма кооперації (рис.4.8) рівня прикладів, який був реалізований, показує саме ці зв'язки між класами. На діаграмі відображені тільки релевантні класи, які мають безпосередній вплив на те частину програми, яка буде реалізовуватися.

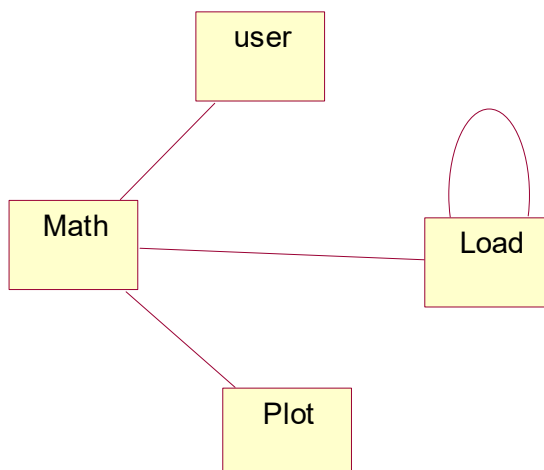


Рисунок 4.8. – Діаграма кооперацій

Діаграма послідовностей та кооперацій є дуже схожими та близькими по своїй логічній суті, але вони мають певні відмінності та сфери застосування. Діаграма послідовностей зосереджена на відображенні саме часових послідовностей передачі повідомлень та коли саме в часі відбувається той чи інший процес.

Діаграма кооперацій здатна показати тільки загальну статичну взаємодію об'єктів, без деталізації. Діаграма послідовності більш важлива для розробників програми, аби точніше розуміти логіку програму, а діаграма кооперацій виступає

як спрощений її варіант, який для розробників несе не дуже багато користі, але такий простіший варіант є набагато зрозуміліший для замовників.

4.9 Діаграма станів

Діаграма станів роботи головного сценарію використовують для показу, як об'єкт системи переходить від одного стану в інший в процесі виконання програми. Зазвичай цю діаграму застосовують з точки розу аналітиків, для більш зрозумілішого пояснення функціонування об'єкта (рис.4.9).

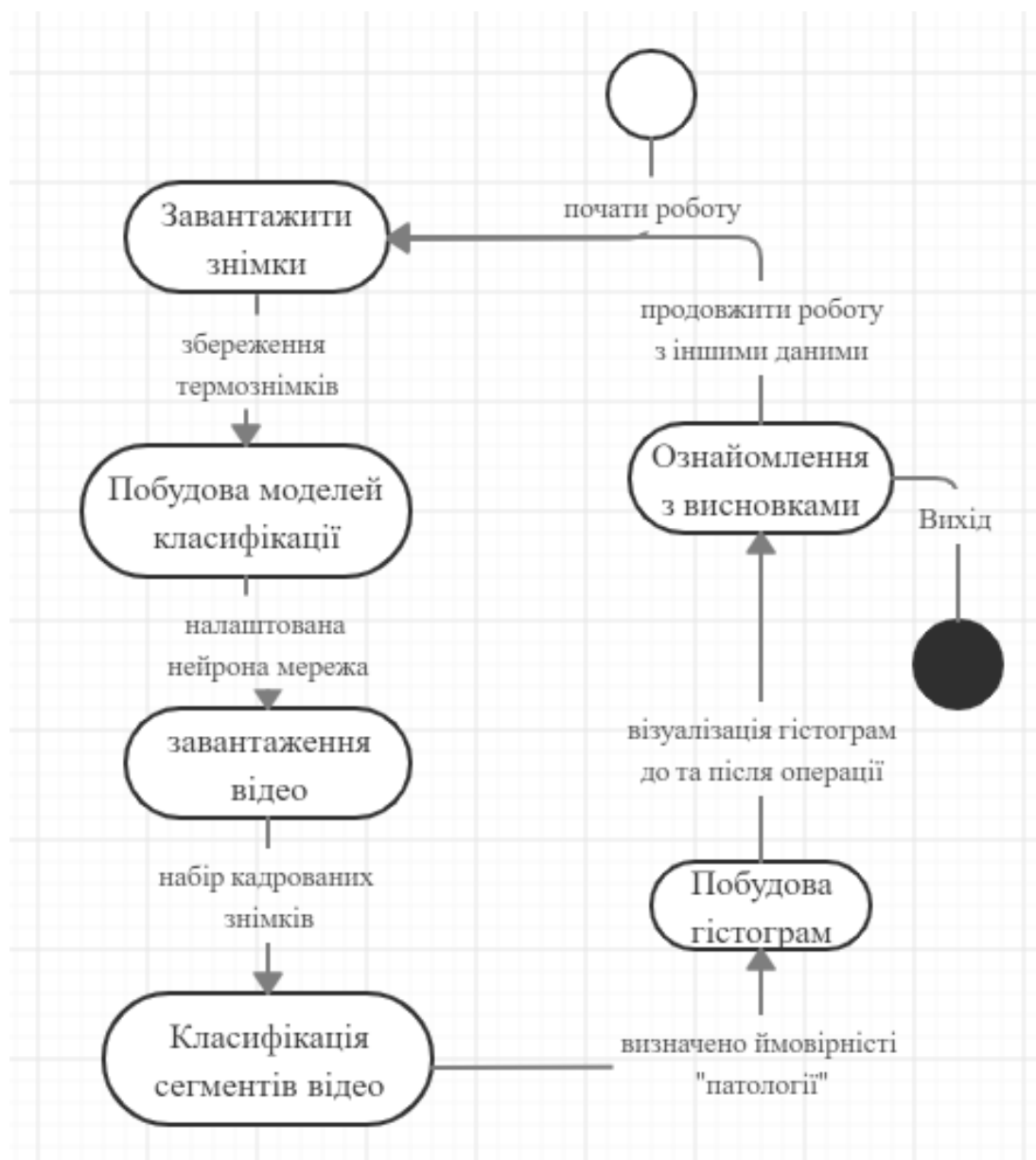


Рисунок 4.9 – Діаграма станів

Основними елементами є «стан» та «перехід», які позначені прямокутником і стрілочкою відповідно. В такій діаграмі прийнято, що дія відбувається миттєво і при цьому об'єкт знаходиться в заданому стані довгий час, що і відрізняє її від інших діаграм.

В діаграмі станів виконується правило, що для переходу з одного стану можна виконати перехід тільки в один стан, наявність умов в таких діаграмах є недопустимою. Таким чином така діаграма підходить для опису конкретного об'єкту, який може приймати декілька станів, але не описують взаємодію з іншими об'єктами.

При знаходженні об'єкта в певному стані, в середині нього відбуваються певні процеси і перехід до наступного стану може відбуватися тільки по завершенню всіх етапів в попередньому, або може ставитися умова виконання деяких умов, які називають сторожові умови. Окрім цього дозволено або об'єкт виходив і заходив сам в себе.

Висновки до розділу 4

В даному розділі, було повністю спроектовано програмний продукт, визначено його основний функціонал, вимоги до нього та зображені всі основні моменти, які важливі для подальшого кодування та реалізації програми. Було визначено необхідні функції, які має включати в себе програма та їх взаємозв'язок один з одним, як з боку користувача так і з боку розробника. На основі створених діаграм в подальшому буде створено програмну реалізацію.

РОЗДІЛ 5

ПРОГРАМНА РЕАЛІЗАЦІЯ

5.1 Вибір мови програмування

На даний час існує багато мов програмування, які широко використовуються для задач машинного навчання як і в комерційній, та і в науковій сфері. Кожна з них має власні переваги та недоліки, та вибір мови залежить в першу чергу від вхідних параметрів та вимог до функціоналу, який розробляється. З основних засобів програмування можна виділити: Python, Matlab, R, Java. Порівняємо їх детальніше за основними критеріями: доступність, наявність необхідних бібліотек, зручність та специфічні переваги і недоліки[44].

Якщо розглядати популярність мов програмування для роботи з даними, то одним з лідерів буде Python. Він має великий функціонал з відкритим кодом, який легко можна включити в будь-який проект або модернізувати під себе. Також Python має широкі можливості не тільки в машинному навчанні, а і в суміжних сферах таких як, веб-розробка, чат-боти, широкий варіант API, що дозволяє обмежитися лише однією мовою програмувати для розробки повноцінного продукту[45].

Також Python має вже готові рішення по нейронним мережам, багато бібліотек, дата сетів, які заточені під конкретну область, такі як, обробка зображень, розпізнання тексту, розуміння мови та інші.

Великою перевагою Python є його універсальність та можливість використання для всіх операційних систем, повна безкоштовність, за винятків специфічних засобів компіляції, можливість працювати як в онлайн режимі так і в десктопному варіанті[46].

Сама мова програмування базується на C++, але при цьому має зручний та гнучкий інтерфейс з великою кількістю специфічних модулів під різні задачі, але при цьому вона не втрачає своєї гнучкості та для складних операційних процесів є

можливість звертатися безпосередньо до мови C++ аби забезпечити обчислювальні здатності на вищих рівнях.

Таким чином, на відміну від високорівневих мов програмування, Python є простим для вивчення та наявність відкритою документації, робить процес освоєння мови зрозумілішим. Також ця мова займає перші позиції, як рекомендована мова для знайомства з програмуванням. Великим плюсом в цій сфері є те, що встановлення відбувається вже через сформовані репозиторії, та одночасно, а додаткове завантаження необхідних бібліотек може відбуватися різними способами, як і через командну строку, так і через графічний інтерфейс.

Також окрім своєї загальної зручності користування, Python має велику кількість специфічних пакетів для машинного та глибокого навчання, які продовжують розвиватися та оптимізуватися, що дає широкі можливості для розробників.

Але не дивлячись на основні переваги та зручність цієї мови програмування в неї є ряд недоліків.

По-перше, якщо її порівнювати з основною мовою, яка заточена саме під роботи зі статистичною обробкою, R, то Python значно програє їй по кількості необхідних бібліотек, але ця ситуація останнім часом поліпшується, адже для Python постійно ведеться розробка нового функціоналу. Таким чином, велика кількість аналітиків, відмовляються від використання великої кількості різних програм під різні задачі, віддаючи перевагу саме Python[47].

Але при цьому Python програє по швидкості роботі своїм конкурентам. Тому при оптимізації процесів на це необхідно звертати особливу уваги та завчасно виділяти, які характеристики є важливими до продукту, який розробляється.

Та основна проблема цієї мови базується та типізації. По-перше, Python має динамічну типізацію, за виступає як і плюсом так і мінусом. Чим більша та масивніша програма, тим стає складніше в ній шукати помилки, які пов'язані з привласненням одних даних до інших. Та при роботі з різними типами даних, масивами, строками, числами, треба уважно слідкувати аби змінні одного типу даних прирівнювалися та такого ж саме типу даних. Часто такі помилки виникають

при роботі з функціями, та копілятор не завжди здатен їх відслідкувати. Але через наявність подібної динамічної типізації, Python компенсує свою повільність роботи та саме цим здатен прискорити процес обробки інформації.

Сумуючи вище сказане, Python повністю підходить під цілі наукової сфери та дозволяє працювати в гнучкому режимі і як виконувати допоміжні розрахунки і для написання повноцінних програм

Іншим популярним засобом для проведення розрахунків в науковій сфері є MATLAB. Він виступає не як повноцінна мова програмування, а як засіб вирішення аналітичних проблем, бо не має наскільки великого функціоналу, як звичайні мови програмування, але при цьому має широкі можливості в сфері аналітиці.

Окрім цього MATLAB є популярним засобом для роботи в наукових дослідках завдяки наявності реалізації складної математичної функціональності.

MATLAB вимагає офіційної ліцензії для роботи, при цьому вартість ліцензії залежить від повноти функціоналу. Це створює проблему для використання його для аналізу даних, адже не всі науково-освітні заклади забезпечують працівників ліцензійними програмами.

Також до недоліків відноситься те, що MATLAB не використовують як самостійний засіб. Зазвичай його комбінують з іншими мовами програмування для отримання повноцінних аналітичних продуктів з інтерфейсами, тому його використання вимагає від аналітика досконалих знань різних засобів програмування, що не завжди є доцільно, адже існує багато інших варіантів, які поєднують в собі всі необхідні функції.

Але не дивлячись на подібні недоліки, MATLAB є гарним інструментом для роботи з числами. Йому надають перевагу при роботі під час кількісного аналізу, особливо при роботі з сигналами, різними типами перетворень Фур'є, вейвлет-аналізу, роботи з великими матричними операціями. Часто це використовується в різних типах інженерії, як простий та швидкий спосіб для проведення специфічних розрахунків, таким чином MATLAB виступає як програмований інженерний калькулятор з розширеним функціоналом, який користувач час створює під свої

задачі. Окрім цього, MATLAB має багато можливостей для роботи з візуалізацією даних, для полегшення сприйняття висновків стосовно аналізу.

Підводячи підсумки по MATLAB, можна сказати, що він є гарною мовою для реалізації математичних функцій, але не для створення повноцінних програмних додатків з великим функціоналом.

Java широкофункціональна мова програмування та має ряд проектів, які заточені для роботи з даними, такими як машинне навчання, нейро-лінгвістичне програмування, графі. Вона відноситься до об'єктно-орієнтованої мови програмування та надає широкі можливості користувачам. Це є високопродуктивна мова, що дозволяє швидко та ефективно працювати з запитами завантаження-перетворення даних, а також дозволяє оптимально використовувати ресурси комп'ютера під час роботи з великими даними.

Java після 8 версії є безкоштовною мовою програмування.

Особливість цієї мови є строга типізація даних, що дозволяє забезпечити безпеку при розробці великих додатків та під час роботи з великими даними. Таким чином зникає повністю можливість прирівняти строковий тип даних до числового, що забезпечує полегшення в пошуку помилок та додаткових проблем які можуть виникати під час динамічної типізації.

Багато програм для обробки наукових даних створено на мові програмування Java. Вона є універсальною мовою програмування для будь-яких проектів. Вона забезпечує велику функціональність та можливість інтеграції у вже існуючі системи.

До її недоліків в основному відносять багатомовність. Аби написати простий ввід або вивід інформації знадобиться порядку п'яти строк, що в інших мовах програмування можна зробити за одну. Таким чином, код здається перевантажений інформацією, яка не несе в собі інформативності, а слугує лише для налаштування всіх необхідних модулів для коректного відображення інформації. Це для початкового користувача може стати проблемою під час вивчення мови, особливо під час переходу з більш лексично простих мов. Також це може ускладнювати читання коду.

Ще одним недоліком, який стосується безпосередньо для під час роботи з даними, є наявність недостатньої кількості необхідних бібліотек, та особливо новітніх статистичних методів. Зазвичай розробнику необхідно з нуля розробляти потрібні функції, для роботи, або створювати свої власні класи, для постійної роботи з ними.

Таким чином, Java займає топові позиції для роботи з даними. Але не дивлячись на всі його переваги стосовно легкої інтеграції і базу коду та легкі маніпуляції з нею, що є великим плюсом при роботі над великим проектом, який складається в декількох частин та має бути постійно оновлюваним. Але через недостатньо розвинутий базовий функціонал, використання Java не є доцільним при використанні одиночних проєктів, бо більшість часу буде уходити на створення та розробку аналітичного функціоналу. В таких випадках ефективніше обирати інші мови програмування з загальними базовими шаблонами основних функцій, які можна легко переформатувати під необхідні вимоги.

Мова програмування R на сьогоднішній день є популярною серед аналітиків даних, адже вона має широкий функціонал бібліотек, які заточені спеціально під вимоги та проблеми роботи з даними. Він також часто використовується в науковій сфері, для складних розрахунків та створення окремих модулів програм, але при цьому створити повноцінний програмний продукт на ньому буде проблематично, адже прийдеться застосовувати інші мови програмування.

R є відкритою та безкоштовною мовою програмування.

З основних її особливостей виділяється повноцінна кількість бібліотек з відкритим кодом, яка реалізує велику кількість методів як для статистичного, так і для числового аналізу. Також ця база постійно поповнюється та при необхідності розробник може створити свої власні пакети. Наразі там широкий вибір для робіт з нейронними мережами, філогенетикою, графами, різними видами регресії та багато іншого.

Окрім цього, користувачу при інсталяції мови програмування, не необхідно виконувати пошук додаткових налаштувань. При базових встановленнях існує можливість одразу встановити розширений функціонал, або обрати лише ті

функції, які є необхідні, при цьому залишаючи можливість до встановлювати це при необхідності.

І якщо казати про аналіз даних, то R має великий ряд бібліотек для їх візуалізації, які мають унікальні можливості та графічні рішення. Таким чином, при побудові графіків та діаграм є можливість обирати стиль візуалізації та мати можливість повністю модернізувати та змінювати полотно.

До недоліків можна віднести специфічність мови програмування. Вона є доволі вузько направленою та дозволяє отримувати гарні результати для статистичних дослідів та аналізів даних, але не є повноцінним засобом для розробки програмних продуктів.

Також R є доволі специфічною мовою програмування. Якщо користувач знає інші мови програмування, то йому буде важко переключитися на роботу в R, тому що вона має ряд специфічних налаштувань стандартних функції, таких як присвоєння, індексація та інші. Це може призводити до великою кількості неявних помилок.

Окрім цього, через те, що сама мова програмування застосовується людьми з різних сфер, які зазвичай далекі від класичного програмування, то читабельність чужого коду значно подає і можливість використовувати бібліотеки, які були розроблені іншими користувача не є ефективною.

Часом до недоліків відносять надмірність бібліотек. Через можливість широкого функціоналу, у користувачів є багато шляхів для вирішення однієї і тієї задачі, та синтаксис рішення може бути не очевидним та вимагати знання надмірної кількості можливої функціональності мови. При цьому непопулярні бібліотеки мають погано описану документацію, що ще більше ускладнює їх використання.

Таким чином, R так само як MATLAB є гарним рішенням для локальних математичних задач. Але при цьому R має велику базу бібліотек, яка постійно оновлюється та повністю відповідає вимогам для наукової сфери. Він є гарним для першочергових цілей, тестування, аналітики, виборі методів аналізу, але не для створення повноцінного програмного продукту[48].

Вивчивши та проаналізувавши п'ять мов програмування, які частіше за все використовуються для роботи з даними та зіставивши це з задачами, які ставляться до програмного продукту, який розробляється, було обрано Python як основну мову програмування. Він має всі необхідні бібліотеки, які повноцінно здатні реалізувати всі вимоги до майбутньої програми, також Python має можливості працювати в двох режимах онлайн та офлайн, з можливістю до підключення до серверів, що необхідно при роботі з нейронними мережами для розвантаження ноутбука, якщо йому не вистачає власних потужностей.

5.2 Вибір середовища розробки

Під час розробки програмного продукту, виникає необхідність поєднувати декілька середовищ розробки, аби пришвидшити та оптимізувати її.

Як було зазначено раніше, для розробки програмного продукту було обрано мову програмування Python. Для цієї мови існує багато платформ та сервісів, які дозволяють запускати код. Як основний було використано Anaconda. Це безкоштовна платформа, яка може встановлюватися на всі операційні системи та забезпечує доступ до багатьох стандартних пакетів Python, а також можливість легко завантажувати додаткові бібліотек[49].

Завдяки своїй будові, Anaconda дозволяє позбавитися від додаткових налаштувань та завантажень через рір. Це допомагає вирішити основні питання по налаштуванню середовища одразу з встановленням програмного забезпечення та спрощує його редагування вже в процесі розробки. Але при цьому, при необхідності роботи через командну строку залишається можливість працювати через conda, як з менеджером пакетів, який має аналогічний функціонал як і рір.

Також під час розробки та тренування нейронної мережі були використані додаткові потужності сторонніх серверів, адже характеристика ноутбуку не дозволяли ефективно та швидко робити ці процеси. Для цього було використано Google Colab.

Це безкоштовний хмарний сервіс від Google, який дозволяє працювати зі швидкими GPU и TPU та він надає широкий функціонал для машинного навчання, а також базується на Jupyter Notebook, що полегшує роботу з кодом та дозволяє запускати програму частинами. Це надає можливості для швидкого тестування та знаходження помилок. Але при цьому має бути стабільне інтернет підключення, бо це є браузерною сторінкою[50].

З основних переваг його використання виділяють наступні:

Інтеграція з GitHub. Це дозволяє швидко та без проблем створювати, завантажувати та зберігати всі зміни в коді безпосередньо на репозиторії GitHub, який широко використовується під час розробки.

Також Google Colab має просту синхронізацію з Google диском, та при необхідності роботи з даними, їх можна завантажувати безпосередньо з хмарного середовища, що значно пришвидшує процес та дозволяє повністю не прив'язуватися до фізичної машини.

Окрім цього Google Colab не потребує додаткових налаштувань чи встановлень окрім браузера. При переході на сайт та створенні нового документу можна одразу починати роботу з даними.

З цього випливає і те, що Google Colab буде працювати в не залежності від операційної системи на комп'ютері та не буде мати ніякої різниці в інтерфейсі чи функціоналі, при зміні операційної системи.

І основною перевагою є безкоштовність використання Google Colab для тренування моделей, на відміну від аналогічних ресурсів. Саме через це зазвичай і обирають даний сервіс, бо звичайні стаціонарні машини не призначені та не мають настільки великої потужності для обробки великих масивів інформації, які зазвичай використовуються під в машинному навчанні.

Також додатковою програмою для полегшення створення інтерфейса програмного додатку було задіяно Qt Designer на основі PyQt5.

PyQt – це графічний фреймворк на основі Qt, який спеціально розроблявся під Python та дозволяє візуально створювати інтерфейс та потім його інтегрувати в тіло програми.

PyQt як і інші програми, які були задіяні під час розробки, є кросплатформенним та існує дві версії 4 та 5. Та повністю відповідають функціоналу Qt відповідного покоління.

За допомогою Qt Designer генерується графічна складова програми, а потім через руіс відбувається генерація коду безпосередньо для Python.

В графічному дизайнері є широкий функціонал, який налічує близько 6000 класів, методів та функції серед яких:

- доступ до популярних баз даних MySQL, PostgreSQL, Oracle;
- парсер XML;
- інтеграція з HTML;
- підтримка відео та аудіо форматів;
- підтримка багатьох мов функціонування;

Таким чином PyQt забезпечує всі необхідні функції для роботи з програмними додатками та при необхідності, розробник здатен сам доопрацювати та модернізувати код, модернізувавши його необхідним чином[51].

5.3 Порівняння бібліотек для машинного навчання

Для полегшення роботи з машинним навчанням існує багато бібліотек, кожна з яких має певні специфіки та функціонал. Взагалі подібні фреймворки можна розділити на три основні групи.

До першої групи входять бібліотеки так звані конструктори. Основна особливість яких закладається в тому, що користувач має певну кількість готових блоків і сам з них складає граф обчислень та запускає його. Але при цьому, створення нових та унікальних блоків може бути проблематичним, адже потребує специфічних знань з програмування. Таким чином повністю модифікувати такі системи під власні потреби буде сильно проблематично, але при цьому якщо використовувати тільки готові блоки, то можна досягти великої швидкості в розробках, адже весь код вже є оптимізований та відлагоджений для конкретних цілей. Тому обираючи такий вид фреймворків необхідно проаналізувати та

спроєктувати весь необхідний функціонал, який буде використовуватися, і якщо подібні системи здатні його забезпечити, то вони стануть ідеальними для вишішення цих проблем. До представників даного виду можна віднести такі системи, як Caffe, Kaldi, Keras.

Фреймворки які використовують статичний граф обчислення. Основна відмінність від попереднього типу є те, що тепер в розробника є можливість робити будь-які маніпуляції з графом. Розробник сам проектує мережу з нуля, додаючи всі необхідні функції, але після компіляції програми зміни стають неможливими. У користувача лишається можливість тільки для запуску програми, без можливості зміни. Таким чином, подібний підхід поєднує в собі одночасно велику гнучкість під час розробки і забезпечує швидке виконання на етапі роботи з програмою. Але виявлення помилок чи якісь зміни потребують більших зусиль. Іноді для модернізації та налагодження уходить більше час ніж на сам процес програмування, що частково нагадує логіку функціональних мов програмування. Найбільш відомими представниками є TensorFlow, MXNet.

Останнім типом є динамічний граф обчислень. Це модернізований статичний граф, але окрім його гнучкості під час проектування, динамічний граф зберігає гнучкість і після компіляції на момент модернізації коду. Якщо розглядати детально, то в такому випадку окремого графу не існує, він виступає занадто складною системою і його побудова не є можливою, тому він починає існувати тільки в момент використання. Тобто, будується кожного разу в процесі запуску програми для прямого та зворотного проходу, а потім зникає. Саме це і забезпечує максимальну гнучкість та оптимальність роботи на всіх етапах розробки, та дозволяє необмежено використовувати властивості мови програмування, при цьому нічим не обмежуючи розробників. До таких фреймворків можна віднести Torch и PyTorch.

Наразі існує багато бібліотек для роботи з нейронними мережами та машинним навчання, які продовжують розвиватися та модернізуватися, тому користувач має широкий вибір, який залежить виключно від задач проекту та від

власних вподобань розробника. Загальну популярність фреймворків можна спостерігати на рисунку 5.1.

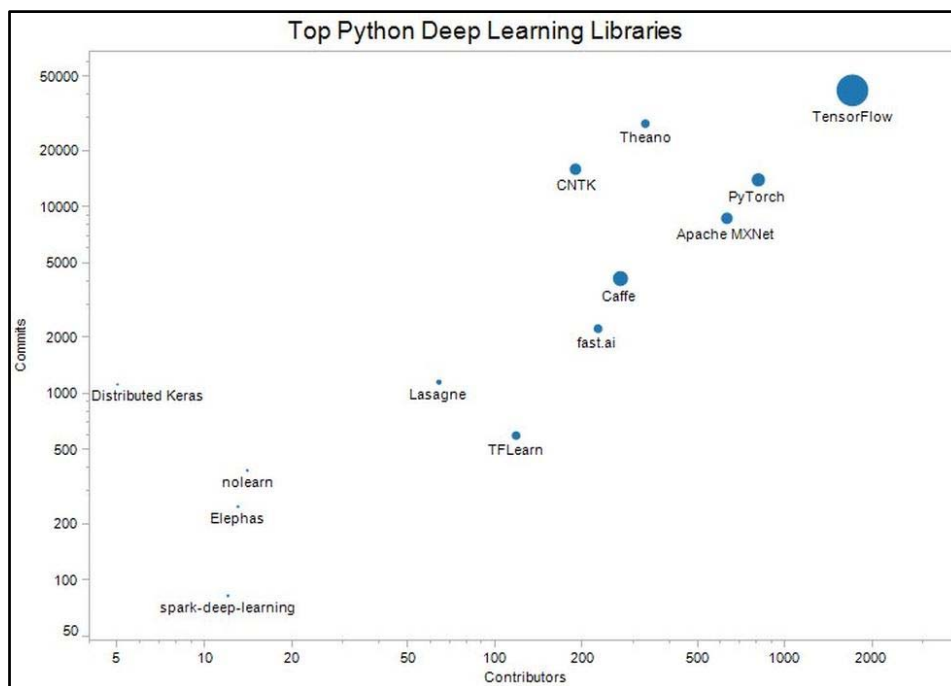


Рисунок 5.1 – Популярність бібліотек машинного навчання

Подивимося на характеристики самих популярних фреймворків. Tensorflow це бібліотека від Google, яка наразі є самою популярною, адже вона призначена для вирішення складних задач, та її використовують великі корпорації.

Цей фреймворк здатен вирішувати великі задачі, будувати багатошарові мережі та гарно справляється з завданнями аналізу картинок, навчання перекладу тексту та розпізнавання голосу, що Google активно використовує в своїх продуктах.

З основних переваг Tensorflow зазвичай виділяють наступні моменти:

- підтримка роботи з мобільними пристроями;
- технічна підтримка розробників та постійна модернізація;
- повноцінна документація для фреймворка;
- засоби спостереження за процесом навчання моделей та проста візуалізація.

До мінусів можна віднести наступні пункти:

- має більший поріг входу для розробників в порівнянні з іншими аналогами;

- процес знаходження на налагодження помилок сильно ускладнений;
- використання всього функціоналу можливо тільки для Python.

Наступною популярною бібліотекою є PyTorch від Facebook, які виступають основним конкурентом до TensorFlow, та використовується у багатьох власних додатках Facebook, університетах та комерційних компаніях.

Основне призначення – швидка, ефективна розробка та навчання моделей, саме це ставили як базову задачу для PyTorch.

З основних його переваг є:

- простий процес створення моделей;
- декларативний паралелізм даних;
- в базі міститься багато готових моделей та даних, на яких користувач може ознайомитися з процесами побудова мереж та при необхідності змінити їх під свої аналогічні потреби.

Основними мінусами виступають недосконалість фреймворку, який є доволі новим та певні функції інтерфейсу є ще відсутніми, такі як наглядна візуалізація та моніторинг процесу навчання, недостатня підтримка моделей, але з кожним новим оновленням подібних проблем стає все менше та всі вони є не критичними для розробника.

Останньою бібліотекою, яка має велику популярність є Keras, яка може працювати як самостійно, так і комбінуватися з іншими фреймворками. В момент розробки основна функція полягала в можливості надати більше швидкості для проведення різних експериментів, аби надати можливість більш швидкого тестування різних методів. Наразі його часто використовують для задач перекладу, аналізу тексту, голосу.

Основні переваги наступні:

- має низький поріг входу та є інтуїтивно зрозумілим;
- підтримує навчання на декількох GPU;
- повністю конфігуруванні моделі;
- не важкий для побудови мереж глибокого навчання;
- підтримує GPU від NVIDIA та TPU від Google.

Але є ряд недоліків:

- малий функціонал в порівнянні з іншими фреймворками;
- не має великої гнучкості для створення специфічних моделей;
- має обмеження, встановленні Tensorflow, CNTK и Theano.

Таким чином можна побачити велику перевагу для PyTorch. Багато аналітиків передбачають великий ріст цього фреймворку і що його почнуть підтримувати більше розробників і з часом він може не тільки зрівнятися з TensorFlow з популярності, а і претендувати на його місце лідера.

Якщо дивитися загальний рейтинг, то наразі кількість розробників, які користуються цими фреймворками однакова, та при цьому спільнота TensorFlow є набагато більшою та стабільнішою, адже він з'явився набагато раніше та завоював довіру користувачів, але PyTorch починає активно завойовувати науково-дослідницьку сферу, його часто використовують при аналізі даних після експериментів та все більше університетів включають його в свої курси з програмування та аналізу даних. Таким чином з часом відбудуватиметься значний зріс користувачів цього фреймворку. Також використання в науковій сфері дає PyTorch репутацію точного та надійного методи побудови моделей, адже при аналізі експериментальних велика увага приділяється саме цим характеристикам.

Таким чином PyTorch є не тільки інтуїтивно зрозумілий фреймворк для аналізу даних, а і представляє з себе повноцінну бібліотеку для розробки в Python, коли TensorFlow виступає як звичайне API та вся взаємодія відбувається через цього, а для заглиблення в програмування розробник повинен мати більш серйозні навички. Також PyTorch, не дивлячись що він з'явився пізніше, в деяких речах переганяє TensorFlow. Таким чином в PyTorch одразу була задіяна система з побудовою динамічних моделей, в той час як у TensorFlow ця можливість з'явилася лише в версії 2.0.

Отже наразі спостерігається великий ріст технологій та їх конкуренцію один з одним. Наразі розроблюється багато локальних фреймворків, які використовують в специфічних сферах, але при цьому кожна бібліотека має свої переваги та недоліки,

на які треба звертати уваги в розрізі поставлених завдань. Нараді для наукової сфери несумнівним фаворитом є PyTorch.

5.4 Опис основних бібліотек

Через те, що візуалізація грає першочергову роль при сприйнятті інформації і є самим простим способом донесення інформації, то необхідно приділяти багато уваги до цієї сторони програмного продукту під час його проектування. А аналітиці візуалізація також відіграє не останню роль та допомагає знаходити закономірності, кореляції, знаходити викиди та артефакти. Такі технології допомагають для подальшої побудови стратегій аналізу або можуть слугувати повноцінним висновком аналізу. Також ці всі методи використовують і в машинному навчанні та багато бібліотек розробляється з прив'язкою до візуального вирішення задачі[52].

Matplotlib є однією з основних бібліотек для візуалізації даних. Вона має стандартний функціонал та нагадує Matlab, при цьому залишаючись безкоштовним та працюючи на Python, що дозволяє робити процес розробки більш простим та інтуїтивно зрозумілим. Тому при переході від стандартної роботи в Matlab, яка розповсюджена в науковій сфері до більш серйозних мов програмування, процес візуалізації залишиться максимально звичним. Окрім цього в Matplotlib можна працювати як з двомірною системою координат, так і з трьох вимірною, що дозволяє вирішувати ширший пласт задач.

Особливістю роботи в Matplotlib є те, що всі об'єкти мають свою ієрархію та представляє з себе деревоподібну структуру.

Figure є одним з головних зовнішнім контейнером для побудови графіки та який може поєднувати в собі декілька осей, які представляють з себе різні види графіків чи діаграм. Таким чином Figure зберігає в середині себе всі інші елементи. В свою чергу осі – Axes – теж має певний ряд підструктур які знаходяться в середині нього. Це можуть бути легенда, текстові об'єкти, мітки та багато іншого.

При цьому кожен з цих елементів також має свої індивідуальні налаштування для роботи з них. Всі основні елементи Matplotlib зображені на рисунку 5.2.

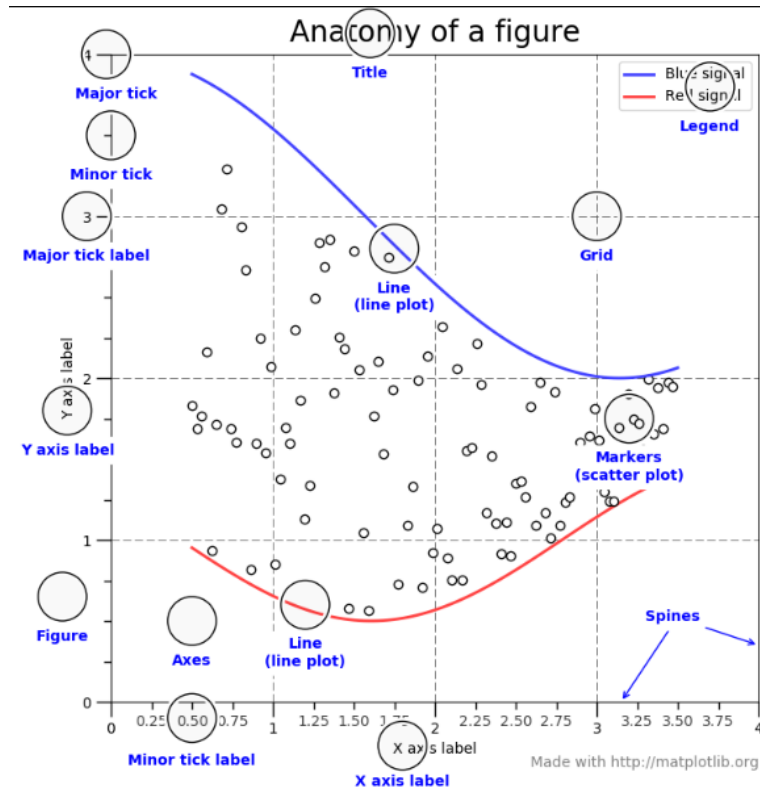


Рисунок 5.2 – Елементи Matplotlib

Основним елементом візуалізації в програмному додатку є стовпчасті діаграми, які будуть відображати ймовірність наявності різкого градієнта температури в певний момент часу.

Таким чином дані діляться на дві категорії, там де ймовірність дорівнює нулю та одиниці, що легко можна буде зрозуміти візуально. Також можливим варіантом є використання гістограм, але через те, що вирішення приймає лише два бінарні значення, то цей розподіл не дасть ніякої важливої інформації, його доцільно використовувати якщо значення є дискретними або їх можна розбити на підгрупи, які будуть включати певні проміжки.

Окрім цього, основним елементом сприйняття є колір, тому доцільно використовувати різну кольорову гаму для різних даних, що дозволяє роботи Matplotlib. Таким чином, використовуючи функціонал бібліотеки, будемо два

окремі графіки зі стовпчастими діаграми, аби розділити дані до операції, під час охолодження, і після, під час нагрівання.

Окрім стандартних бібліотек NumPy для роботи з масивами даних, в яких будуть зберігатися матриці об'єктів, тобто всіх картинок термограм в послідовному порядку, необхідно використовувати додаткові засоби для їх завантаження, які будуть сумісні з іншими бібліотеками.

Для таких цілей було обрано модуль glob, який використовується для пошуку всіх шляхів до файлу та збереження їх в масив. Окрім прямого написання шляху до файлів, він може обирати користувачем інтерактивна. Для цього використовуються спеціальні символи для пошуку певних патернів. Знак "*" використовують, як заміна довільної кількості символів, наприклад, "/usr/Python/*/*". Знак "?" використовується для заміни одного символу, коли необхідно знайти значення по шаблону, при цьому інша частина шляху залишається незмінною. Також при пошуку певних обов'язкових символів їх необхідно заключити в спеціальні дужки [], наприклад, вираз [*] відповідає пошуку "*" в тексті.

Для запиту при поверненні списку путей використовується glob.glob(name), де name – шаблон по якому знаходяться об'єкти який може бути абсолютним, коли є прописаний шлях до конкретного об'єкту, або відносний з використанням спеціальних символів.

5.5 Модель життєвого циклу

Життєвий цикл програмного забезпечення є важливою частиною проектування, бо від нього залежить етапи розробки та час, який необхідно використати від проектування до реалізації та експлуатації продукту. На сьогоднішній день існує декілька моделей, які мають власні переваги та недоліки та використовуються під певні характеристики. Розглянемо дві основні з них: каскадну та спіральну, які є найпопулярнішими.

Каскадна модель уявляє з себе лінійне, послідовне виконання кожного етапу роботи починаючи з планування, закінчуючи інтеграцією. Вважається що ця

модель є застарілою для сучасних розробок, бо при виявленні помилок необхідно повертатися в самий початок для їх усунення, але для невеликих програм чи реалізації певних модулів, це є швидкий та надійний метод.

Основне правило каскадної моделі полягає в тому, що послідовність дій є строго упорядкована та наступний крок можна починати тільки після повного завершення попереднього, після цього повернення та доопрацювання попередніх етапів стає неможливим.

Таким чином, таку модуль можна представити як послідовність наступних кроків: аналіз вимог та розробка технічного завдання, проектування продукту, розробка продукту, тестування та налаштування, експлуатація та можлива наявність ще одного додатковий етап супроводу при необхідності (рис.5.3).

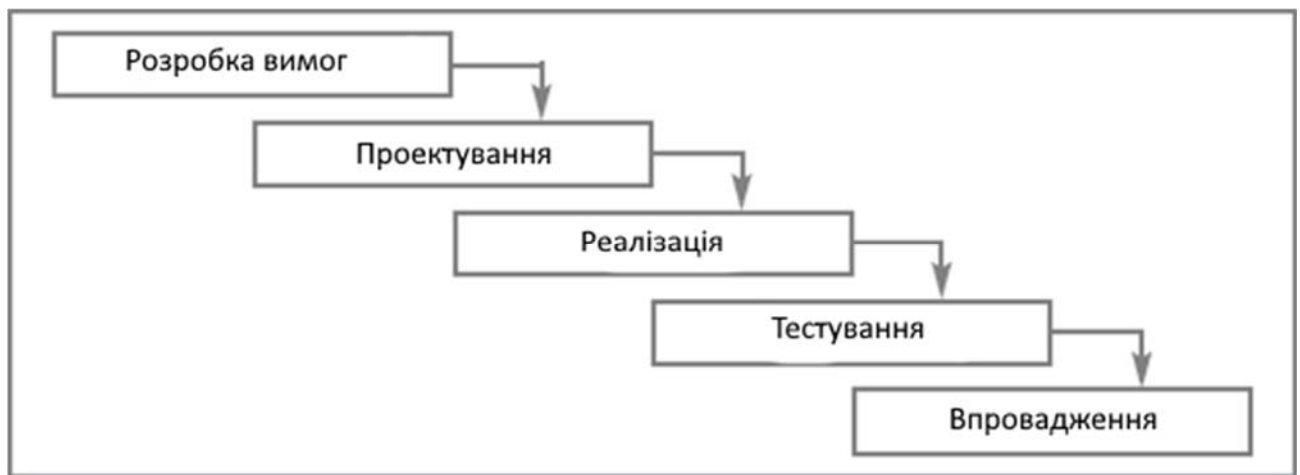


Рисунок 5.3 – Каскадна модель

При цьому подібна модель має ряд переваг:

- неможливість змінити вимоги до технічного завдання та стабільність для розробників в плані постановки задач;
- має велику популярність для реалізації невеликих програм, коли можна на початкових етапах повно та однозначно сформулювати всі вимоги до програмного продукту;
- після кожного етапу розробник має повністю завершений етап з чітко описаною документацією, яка повністю відповідає всім висунутим критеріям та яку використовують для подальшої роботи на наступних етапах;

- повністю зрозуміла логіка послідовності виконання процесів з чіткими строками завершення та розрахунком витрат на реалізацію;
- простота моделі та зрозумілість;
- таким чином, така модель дуже активно використовувалася на початкових етапах розробки програмних продуктів, коли вони не набули своєї масштабності як сьогодні. Через це така каскадна модель має ряд недоліків, через які ця модель втрачає свою популярність:
 - низька гнучкість під час розробки;
 - послідовна структура не дозволяє повертатися на попередні етапи для доробки, бо через це призведе до порушення графіків роботи та затримці результатів, тому необхідно завчасно повністю описувати всі вимоги;
 - складність повно описати вимоги без подальшого їх редагування;
 - користувач бере участь у старенні продукту тільки на етапі розробки вимог та під час тестування, через що немає змоги оцінити проміжну якість продукту;
 - продукт можна використовувати лише коли він буде повністю завершено, немає можливості створити мінімально робочий продукт для тестування та його доповнення;
 - проблеми локальних модулів можна виявити тільки під час зборки продукту, що призведе до додаткових витрат на перероблення продукту;
 - користувачі мають навчатися роботі з програмним додатком вже після його введення в експлуатацію. Вони не мають можливості завчасно з ним ознайомитися та зрозуміти принципи роботи;
 - для новітніх та унікальних систем процес планування, проектування та реалізації може нести великі ризики, адже у такі систем нема аналогів для порівняння і всі вимоги пишуться з нуля, тому існує велика ймовірність не врахувати важливі компоненти.

Таким чином використання цієї моделі має доволі обмежену область використання, адже недоліки мають значну перевагу над перевагами цієї моделі.

Для використання такої моделі в розробці, вихідний продукт має мати наступні характеристики:

- при виконанні великих проектів, де можливо розбиття на окремі модулі, які є прості в своїй реалізації та до яких можна просто описати необхідні, повноцінні вимоги;
- коли відбувається перенос вже існуючої системи на іншу платформу, таким чином завчасно відомі всі вимоги та як має функціонувати продукт, тому для подібних задач не є проблемою лінійність каскадної моделі;
- при оновленні існуючого проекту або написанні нової версії на старій платформі;
- при розробці аналогів системи, які вже реалізовувалися розробниками раніше, та в якому вони знають можливі проблеми та недоліки, які можна прорахувати завчасно.

Отже подібна модель має дуже обмежену область використання та не дає змоги динамічно працювати з проектом, бо будь-які зміни будуть призводити до повної переробки програми, починаючи від вимог. Тому актуальніше буде обрати інший метод проектування, який надасть можливість постійно змінювати систему та модернізувати, адже, зазвичай, вимоги до медичних систем часто змінюють, уточнюються, що вимагає наявності гнучкості при розробці.

Таким чином, зважаючи на всі вище зазначені зауваження до базової моделі, необхідно шукати інші аналоги, які задовільнять цим умовам. Розглянемо каскадну модель життєвого циклу продукту.

Таким аналогом виступає спіральна модель життєвого циклу (рис.5.4), де при кожному витку спіралі виконується нова версія продукту, після цього відбувається уточнення вимог, коректування результатів та планування наступного етапу розробки. Таким чином програмний продукт поступово нарощує все більше функціональних можливостей. Велику увагу приділяють створенню прототипів для кожної з задач. Таким чином кожен наступний етап проектування вже базується на частково готовому продукті, який теж може підлягати змінам.



Рисунок 5.4 – Спиральна модель

Така модель орієнтована на велику увагу до ризиків, які впливають на життєвий цикл продукту, та робить упор на етапах проектування і аналізу, які повторюються під час кожного витка циклу, коли зроблено певну частину продукту та починається робота над іншою частиною.

Після кожного етапу проектування та аналізу розробляється прототип та звіряється з замовником на правильність роботи та перевірку або переробку вимог. Таким чином кожна версія продукту є удосконаленою версією попередньої розробки.

Це дозволяє системі бути максимально гнучкою та точніше підходити до формування вимог, які можуть динамічно змінюватися та доповнюватися, також завдяки цьому можуть змінюватися основні цілі проекту, та на будь-якому етапі після початку розробки проект може повністю змінити свої характеристики. Також це дозволяє більш точно розраховувати час та витрати для розробки, адже подібні розрахунки для невеликих підсистем робляться значно простіше ніж для всього продукту цілком.

Окрім цього для кожного витка розробки системи можуть використовуватися інші методи розробки і навіть каскадна модель. Таким чином основна задача ставиться на те, щоб швидше показати замовнику мінімально життєздатний продукт, отримати від нього правки, зауваження, уточнити вимоги та продовжити розробку. Також в такій системі можливо виконання декількох етапів одночасно, тобто зникає строга лінійність послідовності дій. Але після закінчення кожного циклу розробки має бути життєздатний продукт.

Отже до переваг такої системи можна віднести наступні пункти:

- гнучка модель на всіх етапах;
- в результаті створюється надійна система, адже під час кожного етапу додається новий невеликий модуль програми та це дозволяє повністю протестувати систему та знайти її уразливі місця, аби швидко від них позбутися, бо тестується як і кожен модуль окремо так і вся система цілком та її взаємодія з новими компонентами;
- дозволяє швидко розробити початкову функціональну систему та уточнювати вимоги до неї;
- надає можливість швидко та без додаткових затрат змінювати вимоги до продукту та доповнювати їх;
- користувач приймає активну участь під час створення системи на всіх її етапах, що дозволяє мінімізувати ризики та завчасно виявити недосконалості, та сформулювати критерії для оцінки роботи програми;
- окрім цього замовник має змогу припинити розробку проекту на будь-якому етапі, якщо стає зрозумілим що система буде неефективним засобом для вирішення поставленої проблеми, при цьому це не призведе до втрат великої кількості ресурсів.

Крім цього система все рівно має ряд недоліків, які можуть стати вирішальними під час вибору життєвого циклу. До таких недоліків можна віднести наступне:

- продовження циклів може тривати до нескінченності адже кожного разу замовник може вносити нові права та вимоги, що буде віддаляти момент завершення роботи.

- кожен цикл вимагає створення нової документації, що призводить до її надмірності та розробникам і користувачам буде не зручно з цим працювати;

- для невеликих проектів спіральна модель є занадто ресурс затратною адже розтягує процес реалізації;

- також основна є проблема з визначенням коли саме необхідно переходити на новий цикл та завершити поточний процес. Часом для вирішення вводять додаткові часові обмеження та чіткі плани роботи, але необхідно пам'ятати що завершення попереднього етапу не є строгою умовою для переходу на новий етап, адже можлива паралельна розробка.

Та не дивлячись на ці недоліки наразі спіральна модель є найбільш популярної під час розробки та зазвичай її використовують якщо розробляється нова технологія, в якій неможливо порахувати ризи та однозначно висунути функціональні вимоги, або розробники мають справи з повністю новою для них системою, для мінімізації витрат.

Також якщо завчасно відомо, що вимоги до систему неповні та будуть редагуватися, доцільно використовувати цю модель та проект є довгостроковим, адже зазвичай в таких проектах відбувається багато змін, бо за час створення можуть змінитися не тільки актуальність та ціль програми, а і технічні методи її реалізації.

Окрім цього багато замовників вимагають мати можливість і слідкування за розробкою з можливістю оцінити роботи програми на будь-якому етапі, що потребує створення функціонального продукту за короткий термін.

Отже використання спіральної моделі життєвого циклу є найбільш доцільним методом вирішення для поставлених задач, в першу чергу через те, що вимоги до програмного продукту будуть постійно змінюватися та тестувати аби підібрати оптимальний варіант роботи програми.

5.6 Розробка інтерфейсу

Після того, як всі основні засоби для розробки обрані, можна перейти безпосередньо до цього процесу.

Першим кроком необхідно створити інтерфейс майбутньої програми та визначитися з основними елементами, які він буде включати.

Використовуючи Qt Designer розробляємо шаблон.

На рисунку 5.5 показано початкове вікно роботи з програмою, де з лівої сторони знаходяться всі можливі для використання віджети, з правого боку їх налаштування, а центральна частина представляє з себе полотно, на якому будуть розміщуватися необхідні елементи.

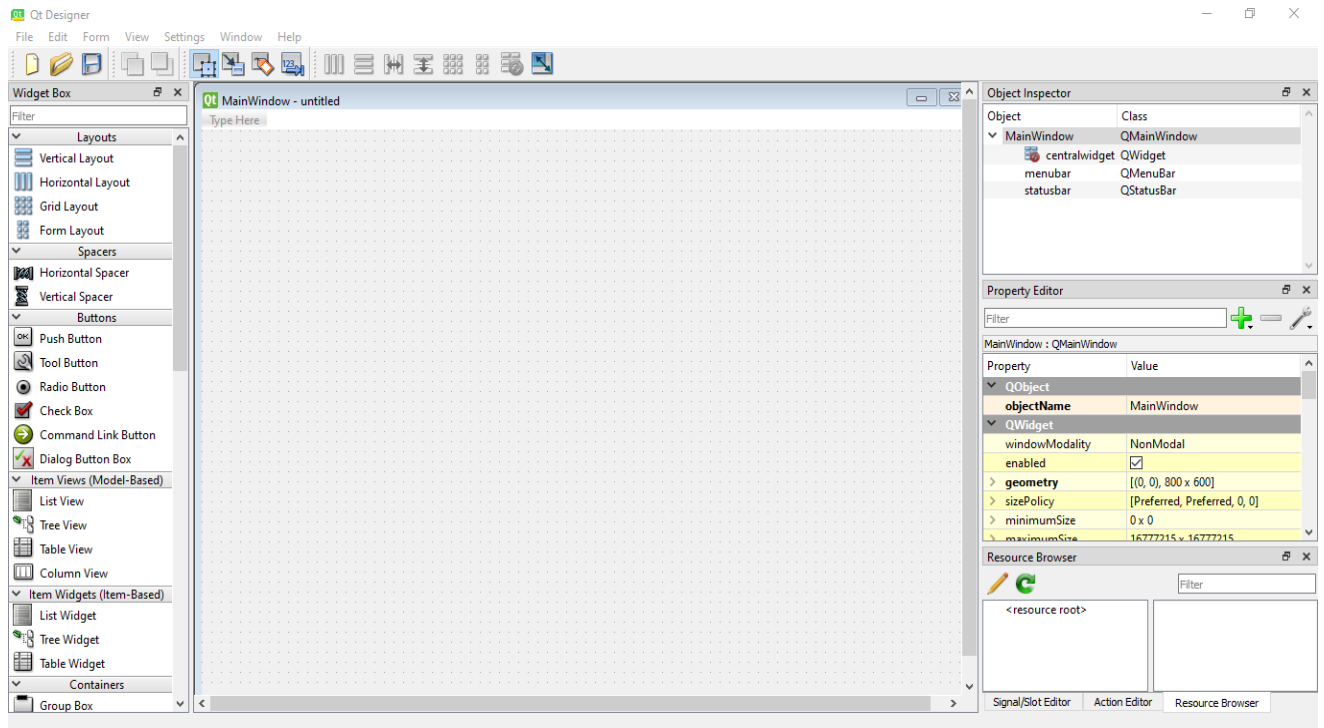


Рисунок 5.5 – Інтерфейс Qt Designer

Визначимо, що необхідно для програмного додатку. По перше, він має надавати можливість для завантаження двох файлів до та після операції, ця взаємодія буде відбуватися за допомогою кнопок.

Далі необхідно створити кнопку запуску програми та також систему сповіщань, яка буде казати на якому етапі роботи знаходить програма, які дії були

завершені успішно, а де сталися певні помилки, таким чином це дає змогу одразу зрозуміти, коли аналіз пішов хибно.

Основним елементом для аналізу, як було сказано раніше, є візуалізація, таким чином необхідно забезпечити місце для побудови двох графіків та також біля кожного з них необхідно вивести основні статистичні показники, на які можна орієнтуватися під час прийняття рішення про успішність операції. За основні показники було обрано три параметри, отже необхідно три поля для виводу.

Останнім елементом є поле з висновками. Це текстовий елемент, на якому буде виводитися інформація на основі порівняння двох графіків, аби користувач бачив одразу висновки та логічні кроки за якими вони були сформовані.

Беручи до уваги всі вищезазначені факти, будемо шаблон інтерфейсу, зображеного на рисунку 5.6.

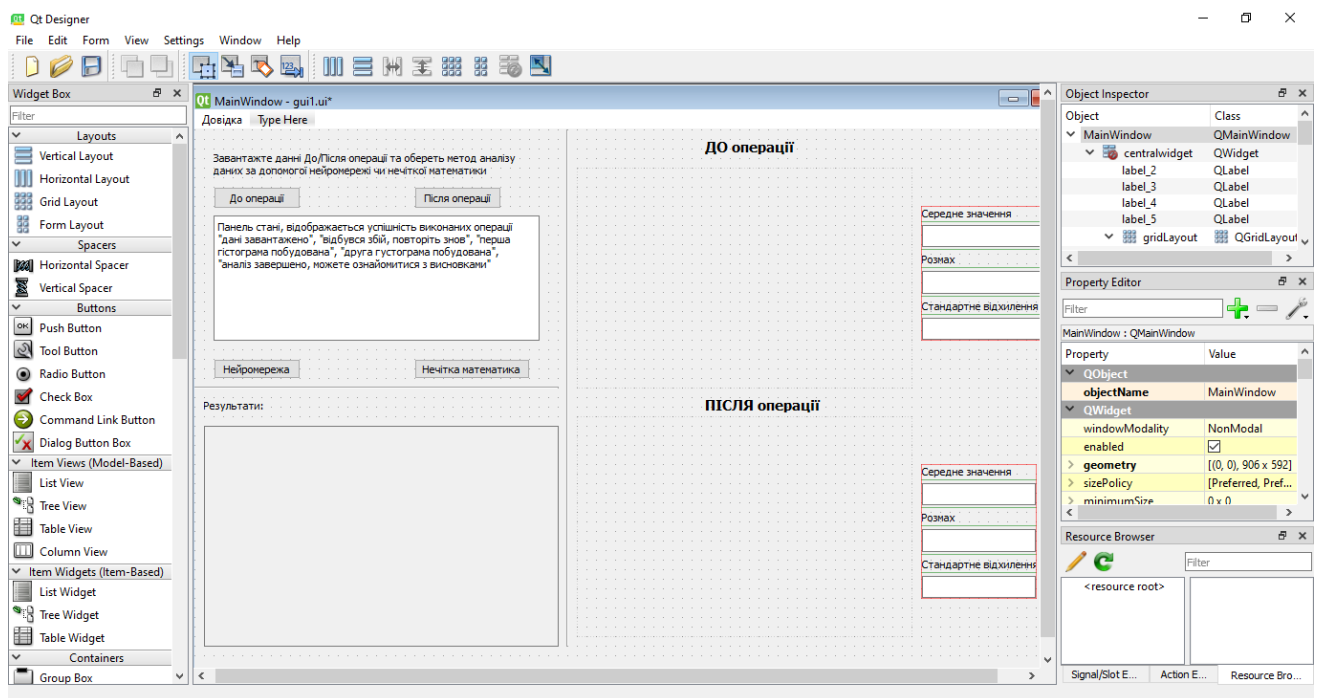


Рисунок 5.6 – Шаблон інтерфейсу

Таким чином, інтерфейс візуально ділиться на три частини: завантаження інформації, візуалізації та висновки, кожна з яких відокремлена від іншої та виконує свої функції, що було зроблено для зручності сприйняття інтерфейсу.

Після того, як шаблон був підготований, його необхідно з'єднати з основною програмою. Для цього необхідно розмістити файл з програмою та файл з інтерфейсом в одному місці, після цього використовуємо спеціальну команду `qtCreatorFile = "gui.ui"`, де `gui.ui` назва файлу з шаблоном інтерфейсу, збереженого з розширенням `.ui`.

Та також необхідно ініціалізувати основні класи для можливої роботи з ним `Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)`.

Після того, як інтерфейс було підключено до програми, необхідно активувати всі елементи з якими взаємодіє користувач, каті як кнопки. Кожна кнопка має своє ім'я `pushButton_*`, де замість `*` прописано її порядковий номер.

Таким чином всі кнопки активуються окремо, та крім цього створюється окремі функції, до яких йде звернення при натисканні, де `function` – назва функції яка виконується. Такий приклад для активації другої кнопки виглядає наступним чином: `self.pushButton_2.clicked.connect(self.function)`.

Наступним етапом є підключення можливості обирати файли з діалогового вікна. Для цього теж існує спеціальна структура в `QFileDialog`, яка має багато елементів, кожен з яких призначений для роботи з певними елементами діалогового вікна. Таким чином, аби надати користувачу можливість інтерактивного вибору шляху, де знаходиться вихідний файл, використовується спеціальні функції.

В цьому випадку вони виглядають наступним чином: `QFileDialog.getOpenFileName(self, "Select a file", "", "All Files (*)", options=options)`, такий запис дозволяє обрати будь-який файл, з неважливо яким розширенням. Шлях до файлу зберігається як строчка та передається до `glob`, який безпосередньо завантажує дані в масив.

Останнім основним елементом інтерфейсу є підключення графіків, для цього теж знадобиться специфічний функціонал. Таким чином спочатку необхідно створити клас полотна на якому буде відображатися графік та всі його елементи. `self.companovka_for_mpl = QVBoxLayout(self.widget)`. Таким чином відбувала ідентифікація простору, до якого маємо доступ.

Окрім цього необхідно окремо створити клас який буде будувати сам графік. Це є власний клас, під час звернення до якого відбувається робота з Matplotlib, цей клас буде мати назву PlotCanvas.

Окрім цього тепер потрібно створити елемент цього класу, через який буде відбуватися взаємодія між елементами: `self.canvasBuild = PlotCanvas(self)`, де `canvasBuild` назва нового елемента класу, який буде передавати дані на полотно з подальшою їх візуалізацією.

Та останнім елементом підключення є надання доступу новому елементу `canvasBuild` до всіх інших елементів класу, для коректного відображення на полотні `self.companovka_for_mpl.addWidget(self.canvasBuild)`.

5.7 Вирішення практичної задачі

Після того, як інтерфейс було підключено до основної програми та було налаштовано всі його основні елементи, можна протестувати програмний додаток та вирішити практичну задачу. Тестування буде відбуватися в двох режимах, безпосередньо використовуючи класифікатор від нейронної мережі та додатково розрахувавши всі показники через математичні функції, аби переконатися в правильності та точності роботи мережі.

Для вирішення практичної задачі візьмем два набори даних під час охолодження серця до операції та під час нагрівання серця після операції, які належать до одного пацієнта, аби порівнюючи дві діаграми температур сказати висновок про результативність операції. Для цього спочатку перетворимо відео на формат зображення та зведемо до однакового шаблонного зразка розмірності, про що описувалася в частині матеріали і методи, аби максимізувати точність роботи класифікатора та надати йому максимально схожі дані, які використовувалися під час навчання.

Запускаємо програмний додаток (рис. 5.7) та бачимо створений раніше інтерфейс.

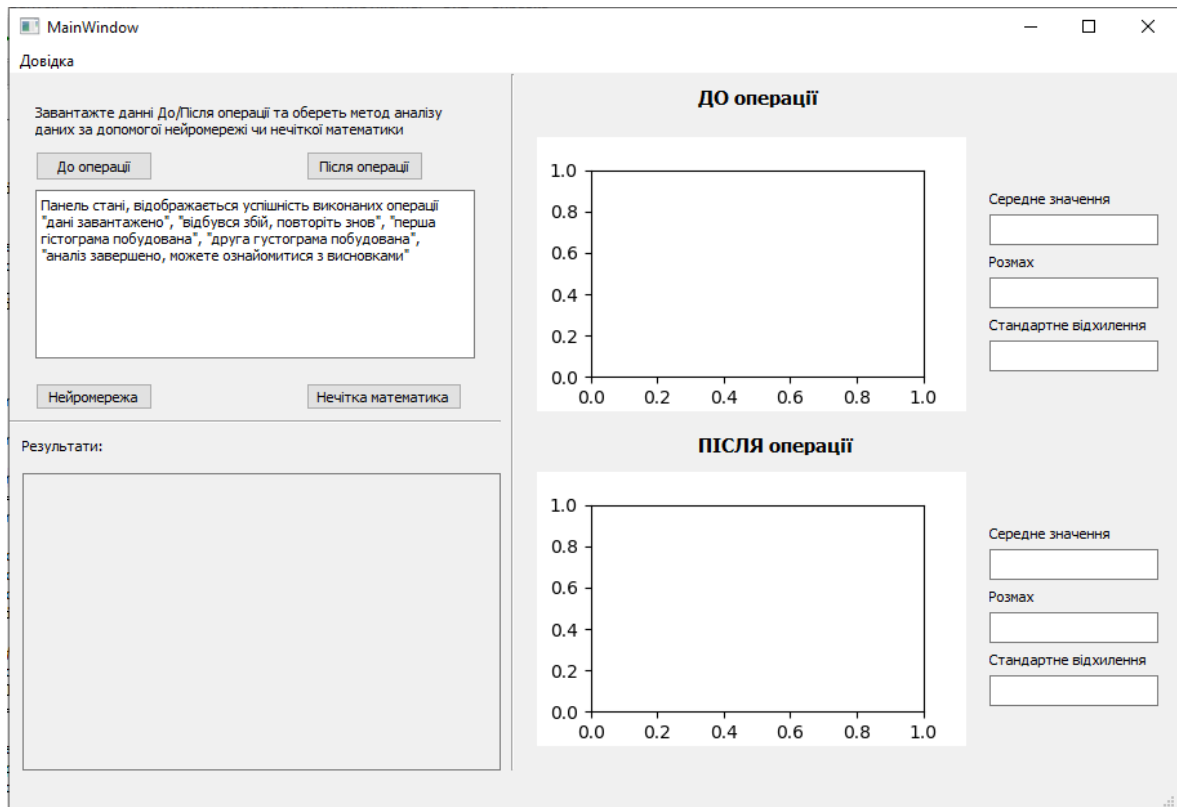


Рисунок 5.7 – Запуск програми

Далі виконуємо інструкції, які написані зверху програми. Завантажуємо через окремі кнопки дані до операції та після операції. В ході завантаження, можна змінювати шлях до папки, при цьому доступ буде здійснюватися по останньому шляху, який було обрано через кнопку, всі інші раніше завантаженні папки не будуть рахувати та братися до уваги. Якщо дана операція завантаження пройшла успішно, то це відобразиться в текстовому табло з даними (рисунок 5.8)

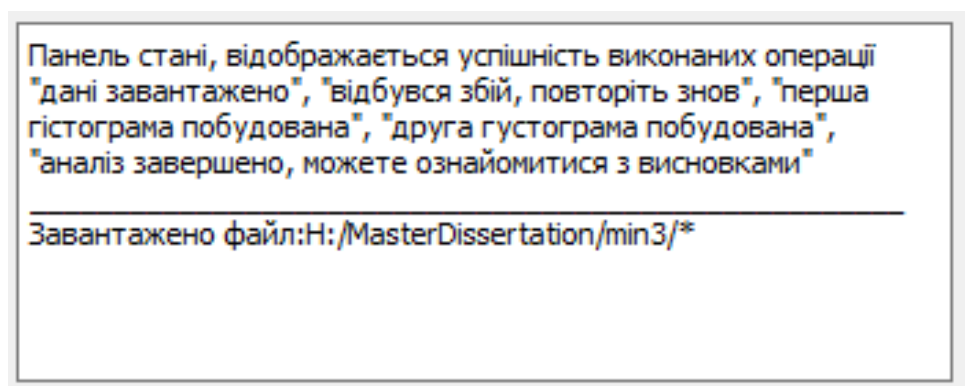


Рисунок 5.8 – Завантаження всіх файлів з папки MasterDissertation/min3.

Наступним етапом є запуск програми з використанням нейронної мережі та ознайомлення з результатами (рис. 5.9).

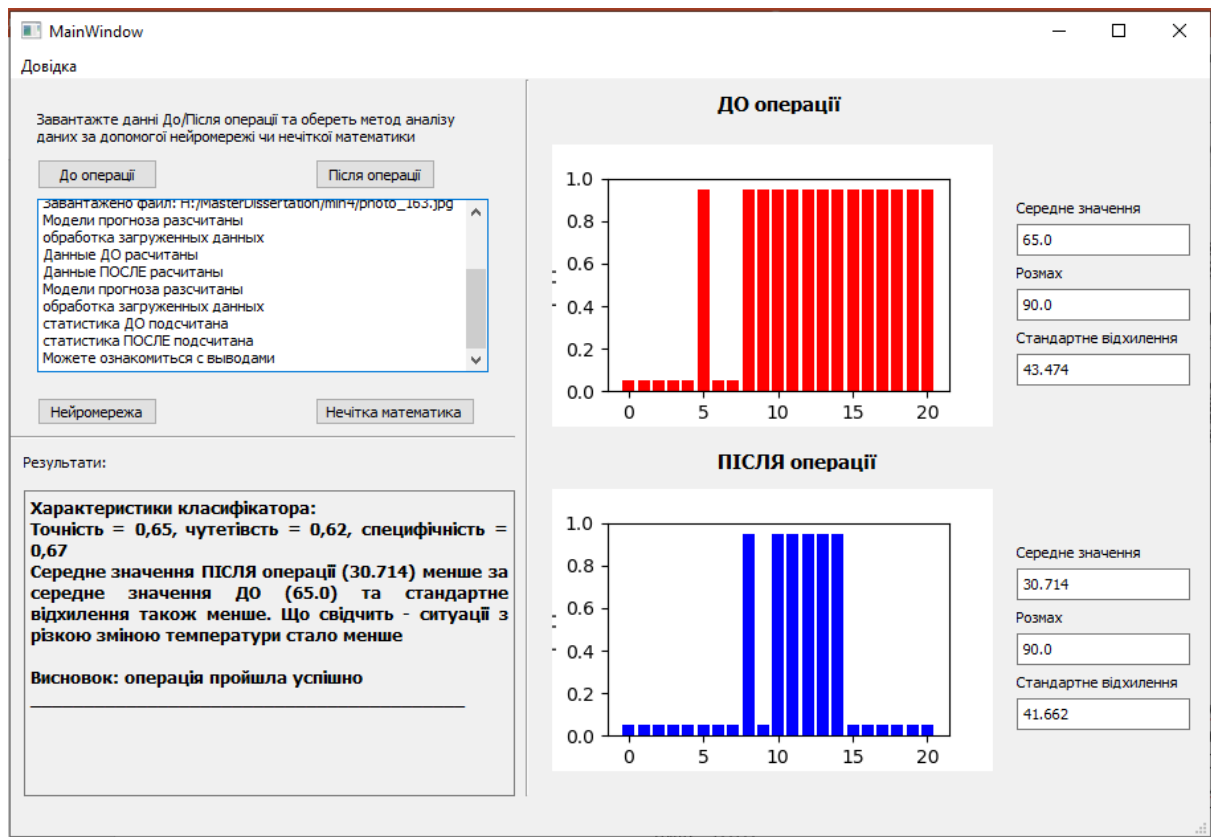


Рисунок 5.9 – Робота програми

Таким чином можемо побачити, що побудувалися два графіки, де червоним зазначено графік до операції, коли серце охолоджувалося, та синім графік після операції під час нагрівання серця, які відображають динаміку зміни градієнтів температури, де низькі піки біля нуля позначають відсутність зон, де градієнт перевищує три градуси, а високі піки біля одиниці позначають наявність зон на знімку з градієнтом більшим за три.

Таким чином, всі знімки розмішені в строгій часовій послідовності та є можливість спостерігати за змінами температури в динаміці, бо поодинокі піки можуть бути свідченням не тільки наявності проблем з кровопостачанням, а і занадто швидким нагрівом чи охолодженням серця, коли температура ще не встигла розподілитися по всьому простору. Таким чином показовими є саме скупчення піків градієнта температури, а не тільки поодинокі значення.

Але через те, що дана система класифікації дає загальну точність класифікації в 65% і для її збільшення необхідно модернізувати алгоритм, який буде перераховувати значення моделі.

Таким чином додатково підключимо математичний розрахунок за допомогою нечіткої логіки, який буде у відсотковій шкалі повідомляти про ймовірність наявності різкої зміни градієнта температури та будувати аналогічну стовпчасту діаграму для кожного моменту в часі.

Цей спосіб є додатковим, але також може використовуватися як основний для аналізу теоремознімків. Головний його недолік в тому, що для обробки математичних функцій уходить значно більше часу ніж для аналізу за допомоги нейронної мережі. Отже таку систему не є можливим використовувати в режимі реального часу, аби одразу після операції сказати про її результативність та при необхідності зробити певні корегування.

Такий спосіб доцільно використовувати тільки як засіб налаштування алгоритму класифікації, що дає змогу зберігати нові дані та накопичувати базу для аналізу і час від часу оновлювати запускати нейронну мережу на основі нової бази даних для визначення нових моделей.

Але аби підвищити ймовірність правильності розпізнання успішності операції, на етапах коли класифікатор ще дає доволі низькі значення, використання подібних додаткових розрахунків є можливим.

Робимо аналогічні дії як в при попередньому запуску програми. Завантажуємо ті ж самі дані та аналізуємо за допомогою математики. Одразу можна побачити, що швидкість роботи програми значно впала та помітні моменти зависання інтерфейсу, що означає, що для використаної обчислювальної техніки не вистачає потужності для швидкої обробки, отже необхідно звернути ретельну увагу на те, на яких пристроях буде запускатися дана програма, адже від їх характеристик на пряму буде залежати здатність до швидкої роботи.

Таким чином було збережено всі основні елементи підрахунку успішності операції і додано аналіз гістограм за критерієм Манна-Уїтні, адже побідні діаграми

важче аналізувати людським оком та визначити наскільки один розподіл значень схожий на інший. З отриманими даними можна ознайомитися на рисунку 5.10.

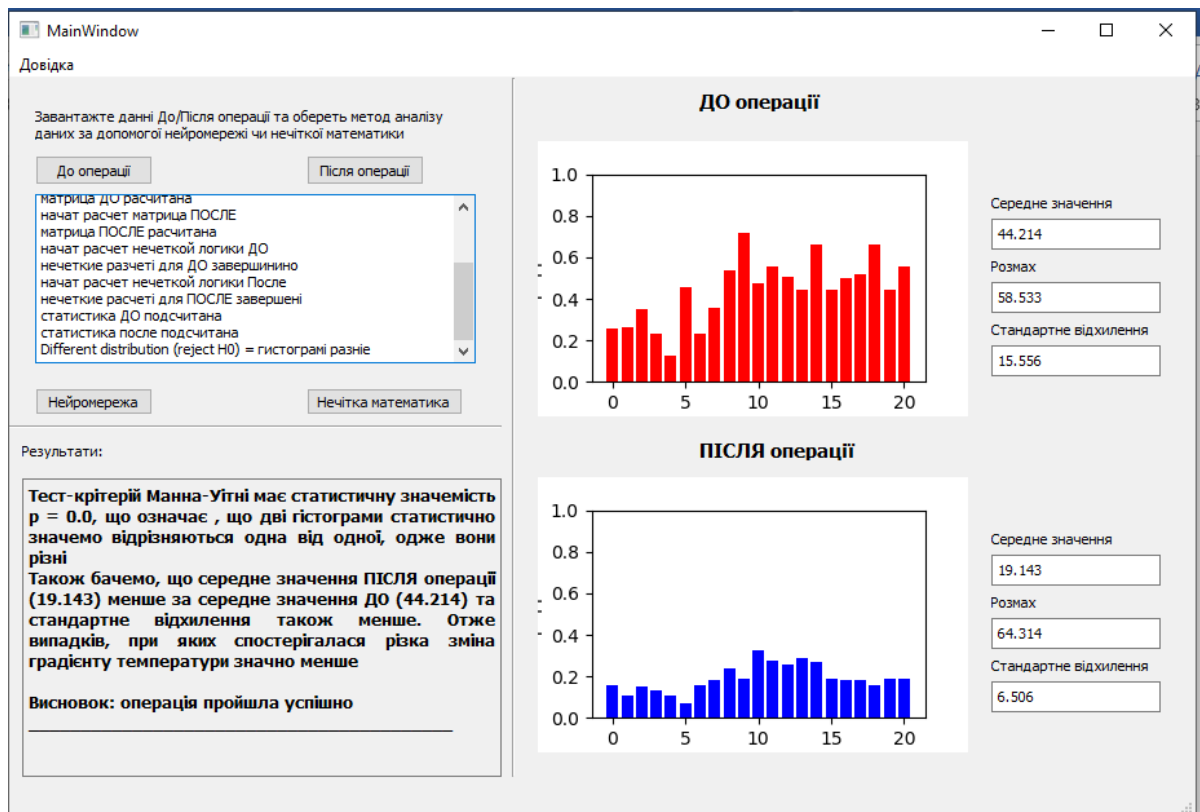


Рисунок 5.10 – Інтерфейс з нечіткою логікою

Такий аналіз дав аналогічні результати як і нейрона мережа, бачимо значне зменшення значення середнього значення та стандартного відхилення та за критерієм Манні-Уїтні робимо висновок, що діаграми є статистично різні. Отже операція пройшла успішно

Висновки до розділу 5

В даному розділі магістерської дисертації було визначено основні технічні засоби, якими реалізовувався програму. Та в підсумку було повністю розроблено, з'єднано всі модулі та протестовано програмний продукт. Також було вирішено практичну задачу з перевірки результативності проведення операції на відкритому серці через ішемічну хворобу.

РОЗДІЛ 6

РОЗРОБКА СТАРТАП ПРОЕКТУ

6.1 Опис ідеї стартап проекту

Проект потрібен для автоматизації обробки термограм серця, що наразі здійснюється лише лікарями і потребує від них до 3-х років навчання. На початкових етапах необхідно завоювати довіру споживачів за допомогою науково-дослідницьких досягнень, що спростить вихід на медичний ринок. Для споживачів головне, аби товар мав специфічність та чуттєвість, які прагнуть до 100%. Таким чином заповнимо дві таблиці з описом проекту (табл.6.1) та визначимо його сильні та слабкі сторони (табл.6.2).

Таблиця 6.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення програми для автоматичного аналізу термограм серця	1. Моніторинг стану серця під час його охолодження або нагрівання	Полегшує роботу для лікарі та робить можливим виявлення всіх небезпечних ситуацій
	2.Визначення результативності операції	Дає змогу кількісно оцінити успішність операції на серці на нормалізацію кровопостачання

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкуренти			
1.	Нейромережа	+	-	Багато часу та трафіку на навчання	доступність	Висока точність
2.	Градiєнтний спуск	+	-	Середня точність	розповсюдження	Довіра користувачів
3.	Вейвлет аналіз	-	+	Довге налаштування	Простий в реалізації	Добре працює на шаблонах

Таким чином було проаналізовано слабкі та сильні сторони продукту та зазначено основну зону направленості проекту.

6.2 Технологічний аудит проекту

Перед тим як братися за ідею необхідно оцінити технічні властивості та можливості і порівняти їх з вже існуючими на ринку аби мати змогу зробити якісний продукт. Проведемо технологічний аудит, за результатами якого заповнимо таблицю 6.3.

Таблиця 6.3 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Нейромережа	Виготовлення ПЗ, написання коду, використання готових архітектур.	Потрібне доопрацювання на навчання на конкретних прикладах	Доступна
2	Градiєнтний спуск	Виготовлення ПЗ, підключення існуючих бібліотек	Наявна	Доступна
3.	Порогова обробка пікселей	Виготовлення ПЗ, написання коду	Потрібна реалізація з нуля	Доступна
4.	Вейвлет аналіз	Виготовлення ПЗ, підключення бібліотек	Потрібне доопрацювання на навчання на конкретних прикладах	Доступна
Обрана технологія реалізації ідеї проекту: нейромережа				

Таким чином для даного продукту є доцільним використання технології на основі нейронної мережі, адже вона найменш ресурс затратна та є доволі гнучкою системою, яку можна модернізувати з часом.

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Наступним етапом є необхідність проаналізувати ринкові загрози та фактори ризиків, аби завчасно мати змоги з ними боротися. Також це допоможе правильно

сформувати пропозицію для потенційних клієнтів та виявити слабкі та сильні місця для конкурентоспроможності.

Визначимо потенційних клієнтів (табл.6.4)

Таблиця 6.4 – Характеристика потенційних клієнтів стартап-проекту

№п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів
1	Автоматична обробка термознімків	Приватні та державні медичні установи.	-	- точність - зручність - безпечність

Після цього є можливість визначити фактори загроз (табл.6.5) та можливостей (табл.6.6) для подальшого розвитку

Таблиця 6.5 – Фактори загроз

№п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Поява аналогічних продуктів на ринку	Прорахувати можливості покращення алгоритму аналізу, розробити систему, яка вимагатиме менше технічного навантаження для пришвидшення роботи.
2	Репутація	В медичній сфері важко отримати довіру від споживача	Випускати наукові статті та працювати на базі наукових закладів задля отримання репутації з надійності та точності результатів.
3	Додаткові потреби користувачів	Продукт не відповідає вимогам користувача	Залишити можливість для покращення та встановлення нових моделей в програмний додаток та збір відгуків від користувачів задля вдосконалення продукту.

Таблиця 6.6 – Фактори можливостей

№п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Інновації	Розробка нових алгоритмів	Покращення продукту за допомогою додавання нового функціоналу та дотримання актуальності для медичних працівників
2	Репутація	Завоювання довіри в медичній сфері для більшої популярності та розвитку	Співпраця з науковими центрами здатна надати гарну репутацію на забезпечити наукове підґрунтя для забезпечення надійного становлення на ринку.

Наступним етапом необхідно виконати більш глибокий аналіз умов конкуренції, для цього була стандартна використана модель М.Портера (табл.6.7), яка здатна охопити всі важливі елементи для аналізу ситуації, визначитися з загрозами від різних сторін.

Таблиця 6.7 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Прямі конкуренти	Бар'єри входження в ринок	Сила постачальників	Сила споживачів	Загрози з боку аналогів
Висновки:	Немає	Бар'єри відсутні, але є ймовірність появи конкурентів	Постачальники не використовують ся	Вимоги до якості та надійності продукту	Аналогів немає

Аналізуючи отримані дані, можна зробити висновок за наразі вихід на ринок з подібним продуктом є актуальним та даний проект бути користуватися попитом.

Не дивлячись на те, що наразі конкурентів у системі не існує необхідно розрахувати фактори конкурентоспроможності та виявити можливі ризики для подальшого розвитку стратегії просування та розповсюдження.

Ці всі основні фактори загроз наведені в таблиці 6.8, в якій проаналізовано шість факторів та наведено підґрунтя для їх використання та визначення переваг чи недоліків.

Таблиця 6.8 – Обґрунтування факторів конкурентоспроможності

№п/п	Фактор конкурентоспроможності	Обґрунтування
1	Ціна	доступний
2	Новий продукт на ринку	На даний час аналогів немає
3	Попит	Розвиток технологій автоматизованого аналізу медичних даних спричиняє збільшення попиту на його виготовлення.
4	Підтримка інфраструктури	Використовуються загальні вимоги під час розробки програмного забезпечення, при цьому відпадає необхідність перероблювати його під замовника та постійно вносити правки, що пришвидшує процес розробки та виходу на ринок
5	Надійність	Система має бути надійна та виконувати всі необхідні засоби щось збереження кондиційності даних
6	Орієнтованість на кінцевого споживача	Продукт орієнтований на взаємодію з клієнтом

Останнім етапом є аналіз на основі слабких та сильних сторін, загроз та можливостей (табл.6.9). Це дасть повноцінне представлення про продукт і змусить подивитися на нього з різних сторін та дати об'єктивну оцінку.

Таблиця 6.9 – SWOT-аналіз проекту.

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> – велика функціональність – сильна команда – інноваційні технології – якісний продукт з високою точністю – немає конкурентів; – точно визначена ЦА 	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> – Легко копіюється – Залежність від політики мед.закладів – Відсутність глобальної реклами – Слабкий імідж товару – Вузька продуктова лінійка
<p>Можливості (O):</p> <ul style="list-style-type: none"> – Зростання ринку автоматизації медицини – Можливість підстроюватися під конкретні вимоги – Впровадження нової технології в сегмент ринку 	<p>Загрози (T):</p> <ul style="list-style-type: none"> – Залежність від лікарів – Нестабільне ціноутворення – Зростання конкуренції – Забезпечення безпеки даних

Таким чином було повністю розроблено та розглянуто аналіз ринкових можливостей та отримано чітку характеристику продукту для подальшої можливості виходу на ринок з мінімізацією ризиків та повним розумінням становлення проекту в загальній сфері.

6.4 Розроблення ринкової стратегії проекту

При розгляданні ринкової стратегії необхідно особливу увагу звернути до вибору цільових груп споживачів (табл.6.10) аби на їх основі уточнити вимоги до системи.

Таблиця 6.10 – Вибір цільових груп потенційних споживачів

№	Сегмент споживачів	Особливості поведінки	Вимоги споживачів	Канали збуту	Інші аспекти взаємовідносин
1.	Медичний персонал в медичних закладах	Споживач має довіру до результатів автоматизованого аналізу та очікує від неї надійної та стабільної поведінки.	Високі вимоги до: - Точність - Чутливість - Специфічність	Прямі канали	Виступи на конференція та наукових зборах для надання підґрунтя актуальності.

Далі на основі вище розрахованих значень необхідно розробити стратегію позиціонування (табл.6.11), яка буде виступати ідентифікатором для споживача.

Таблиця 6.11 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару	Базова стратегія розвитку	Ключові конкурентоспроможні позиції	Вибір асоціацій
	Надійність, точність, зручність	Диференціації	Продукт без аналогів, унікальний алгоритм класифікації	- є інновацією - має наукове підґрунтя - надійність роботи

Таким чином було сформовано стратегію ринкової поведінки проекту.

6.5 Розроблення маркетингової програми стартап-проекту

При побудові маркетингової програми, першим кроком визначаємо переваги товару від прямих конкурентів або ввід майбутніх аналогів для цього товару.

Для більшої наглядної візуалізації, зведемо все до таблиці (табл.6.12).

Таблиця 6.12 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Надійсність	При додатковому збереженні даних, алгоритм може сам перебудувати моделі та самонавчатися.	-
2	Точність та зрозумілість	На виході надається як візуальна складова аналізу та і підраховані статистичні дані	-
3	Автоматичний аналіз	Весь аналіз та постановка результату проводиться автоматично від користувача необхідно тільки завантажити дані.	-

Важливим етапом є формування системи збуту (табл.6.13) та розуміння просування товару на ринку.

Таблиця 6.13 – Формування системи збуту

№п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	До основних клієнтів відносяться лікарі приватних та державних установ, які зацікавлені в впровадженні технологій для діагностики, що значно полегшить для них роботу, тому подібних користувачів можна зустріти на різного типу наукових зборах, таких як конференції.	Встановлення прямих та довірливих стосунків з користувачами та здатність модернізувати продукт під конкретного замовника	Пряма	Споживач безпосередньо контактує з розробниками та вони в свою чергу забезпечують повне встановлення та навчання роботи з системою.

Завершальною задачею в цьому блоці є побудова концепції за якої буде організовуватися маркетингова комунікація, де будуть зазначені основні канали комунікації, обрано позиції компанії, визначено з основними характеристиками, які

будуть доноситися до користувачів для побудови бренду до в коло технології та оцінено поведінку користувачів(табл.6.14).

Таблиця 6.14 – Концепція маркетингових комунікацій

№п/п	Специфіка поведінки клієнтів	Канали комунікацій з клієнтами	Ключові позиції позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	До основних клієнтів відносяться лікарі приватних та державних установ, які зацікавлені в впровадженні технологій для діагностики, що значно полегшить для них роботу, тому подібних користувачів можна зустріти на різного типу наукових зборах, таких як конференції.	Наукові заходи такі як, конференції, семінари та пряма комунікаціями з закладами які працюють з операціями на відкритому серці.	Гнучкість та здатність підлаштуватися до нових споживачів. Не боятися просити зворотній зв'язок.	Зв'язати назву бренду з відчуттям надійності та точності, аби розвивати подібну репутацію, активно будувати позицію через наукову спільноту ті її заохочення подібної технології	Автоматизація даних спрощує життя.

В підсумку маємо повну систему виходу на ринок зі сторони маркетингу та розуміння в подальших кроках просування.

Висновок до розділу 6

В розділі було проаналізовано всі основні елементи для побудови стартап проекту, визначено сильні та слабкі сторони, способи позиціонування на ринку. Таким чином в даних момент часу вихід на ринок з технологією автоматичної обробки термограм є легкою задачею та подальший розвиток є доцільним, адже є цільова аудиторія, якій необхідний аналогічний продукт, а сьгоднішній ринок не має конкурентів в цій сфері.

ВИСНОВКИ

В ході роботи було розглянуто створення системи для динамічного аналізу стану серця за допомогою термограм на основі класифікатора з нейронної мережі.

1. Проведена порівняльна характеристика різних методів класифікації та архітектури нейронних мереж, було обрано оптимальну структуру мережі згортки з 8 шарів для аналізу термограм серця.

2. На основі нейронної мережні згортки було побудовано класифікатор, який дає точність в 65% при класифікації термознімків на ті в яких спостерігається різка зміна градієнта температури більше ніж на три градуси чи ні.

3. Розроблено динамічну систему на основі класифікатора з нейронною мережею, яка дозволяє оцінювати результативність проведеної операції на серці за допомогою порівняння даних до та після операції.

4. Розроблено інтерфейс для зручної взаємодії з програмою, де розписано послідовність дій, які має робити користувач для успішної роботи з додатком.

5. Проаналізовано основні конкурентні алгоритми для автоматичного аналізу зображень та визначено засоби для покращення поточного класифікатора на основі використання нечіткої математики з можливістю перерахунку моделей під час доповнення баз даних пацієнтів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Иваницкий Г. Р. Тепловидение в медицине / Г. Р. Иваницкий // 2006. — Vol. 76, No. 1. — P. 48–58.
2. Скрипаль А. В. Тепловизионная биомедицинская диагностика. / А. В. Скрипаль, А. А. Сагайдачный, Д. А. Усанов // 2009. — P. 118.
3. Блюмин Р. Б. Технологии бесконтактной диагностики / Р. Б. Блюмин, Е. М. Наумова, А. А. Хадарцев // 2008. — Vol. 4, No. 15. — P. 146–149.
4. Эль-Хатиб С. А. Сравнительный анализ алгоритмов выделения контуров медицинских изображений / С. А. Эль-Хатиб, Ю. А. Скобцов // 2011. — No. 52–57.
5. Заяц Г. А. Медицинское тепловидение – современный метод функциональной диагностики / Г. А. Заяц, В. Т. Коваль // 2010. — Vol. 43, No. 3. — P. 27–33.
6. McCullough P. A. Coronary artery disease / P. A. McCullough // Clinical journal of the American Society of Nephrology: CJASN. — 2007. — Vol. 2, No. 3. — P. 611–616.
7. Taylor C. J. Heart failure therapy in patients with coronary artery disease / C. J. Taylor, F. D. R. Hobbs // Current Opinion in Pharmacology. — 2013. — Vol. 13, No. 2. — P. 205–209.
8. Ak M. A review on coronary artery disease, its risk factors, and therapeutics / M. Ak, C. D, H. B, [et al.] // Journal of cellular physiology. — 2019. — Vol. 10, No. 234.
9. A L. The role of coronary artery disease in heart failure / L. A, D. As // Heart failure clinics. — 2014. — Vol. 2, No. 10.
10. Маринина Е. С. Наочное обоснование основных путей профилактики сердечно-сосудистых заболеваний / Е. С. Маринина, О. А. Нагибин // Vol. 2, No. 47. — P. 70–76.

11. Pg S. Future of the prevention and treatment of coronary artery disease / S. Pg, D. G // *Circulation journal : official journal of the Japanese Circulation Society*. — 2016. — Vol. 5, No. 80.
12. Оганов Р. Первичная профилактика ишемической болезни сердца / Р. Оганов // 2009. — Р. 43.
13. Абрамов Н. Информационные системы в медицине / Н. Абрамов. — 2008. — 171 p.
14. Tattersall G. J. Infrared thermography: a non-invasive window into thermal physiology / G. J. Tattersall // *Comparative Biochemistry and Physiology. Part A, Molecular & Integrative Physiology*. — 2016. — Vol. 202. — P. 78–98.
15. Виноградов В. И. Некоторые аспекты применения термографии при реабилитации пациентов с нарушением функций опорно-двигательной и нервной систем / В. И. Виноградов, И. С. Веретенов, В. Н. Сзелко // 2005. — Vol. 3. — P. 76–78.
16. Nowakowski A. Thermographic and electrical measurements for cardiac surgery inspection / A. Nowakowski, M. Kaczmarek, J. Wtorek, J. Siebert // 2001. — P. 2873–2877.
17. Ring E. F. J. Infrared thermal imaging in medicine / E. F. J. Ring, K. Ammer // *Physiological Measurement*. — 2012. — Vol. 33, No. 3. — P. 33–46.
18. P S. Physical activity and arteriosclerotic heart disease / S. P // *Public health reports (Washington, D.C. : 1896)*. — 1955. — Vol. 2, No. 70.
19. В К. A pruned ensemble classifier for effective breast thermogram analysis / К. В, S. G // *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*. — 2013. — Vol. 2013, No. 20.
20. Титова А. Ю. Специализированная компьютерная система обработки термомаммографических изображений / А. Ю. Титова // 2012. — P. 87–92.
21. Caliskan A. Classification of coronary artery disease data sets by using a deep neural network / A. Caliskan, M. E. Yuksel // 2017. — Vol. 1, No. 4. — P. 217–277.

22. Huang C. L. The application of infrared thermography in evaluation of patients at high risk for lower extremity peripheral arterial disease / C. L. Huang, Y. W. Wu, C. L. Hwang // 2011.
23. Ясницкий Л. Н. Нейросетевая система экспресс-диагностики сердечно-сосудистых заболеваний / Л. Н. Ясницкий, А. А. Думлер, А. Н. Полещук, К. В. Богданов // 2011. — Vol. 28, No. 4. — P. 77–86.
24. Kriegeskorte N. Neural network models and deep learning / N. Kriegeskorte, T. Golan // *Current biology: CB.* — 2019. — Vol. 29, No. 7. — P. 231–236.
25. Созыков А. Обзор методов обучения глубоких нейронных сетей / А. Созыков // Vol. 6, No. 3. — P. 28–59.
26. Renganathan V. Overview of artificial neural network models in the biomedical domain / V. Renganathan // *Bratislavske Lekarske Listy.* — 2019. — Vol. 120, No. 7. — P. 536–540.
27. Zayaats G. A. Medical thermal imaging - a modern method of functional diagnostics / G. A. Zayaats, V. T. Koval // *Health. Medical ecolog. Science.* — 2010. — Vol. 43, No. 3. — P. 27–33.
28. S P. Suspicious-region segmentation from breast thermogram using dlpe-based level set method / P. S, B. D, B. D, [et al.] // *IEEE transactions on medical imaging.* — 2019. — Vol. 2, No. 2.
29. Zhang Y. Neural network-based approaches for biomedical relation classification: a review / Y. Zhang, H. Lin, Z. Yang, [et al.] // *Journal of Biomedical Informatics.* — 2019. — Vol. 99. — P. 103–294.
30. LeCun Y. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // *Proceedings of the IEEE.* — 1998. — Vol. 86, No. 11. — P. 2278–2323.
31. Barbosa Pereira C. Contactless monitoring of heart and respiratory rate in anesthetized pigs using infrared thermography / C. Barbosa Pereira, H. Dohmeier, J. Kunczik, [et al.] // *PloS One.* — 2019. — Vol. 14, No. 11. — P. 747.

32. Graaf G. de Heart rate extraction in a headphone using infrared thermometry / G. de Graaf, D. Kuratomi Cruz, J. C. Haartsen, [et al.] // IEEE transactions on biomedical circuits and systems. — 2019. — Vol. 13, No. 5. — P. 1052–1062.
33. Чаша Т. В. Применение нейронных сетей для прогнозирования течения постгипоксических нарушений сердечно-сосудистой системы у новорожденных детей / Т. В. Чаша, Н. В. Харламов, О. И. Климова, Ф. Н. Ясинский // 2009. — No. 4. — P. 57–59.
34. Гірін Р. В. Аналіз метрологічних характеристик тепловізіонних систем технічного діагностування з нейронною мережею / Р. В. Гірін // 2018. — Vol. 4, No. 60. — P. 66–81.
35. Сбоев А. Г. Разработка нейросетевой методики раннего диагностирования ишемической болезни сердца и коронарного атеросклероза / А. Г. Сбоев, С. Г. Горохова, Н. Н. Черный // 2011. — Vol. 2. — P. 2004–2013.
36. Вайнер Б. Г. Матричное тепловидение в физиологии: исследование сосудистых реакций, перспирации и терморегуляции у человека / Б. Г. Вайнер // 2004. — P. 96.
37. Bichinho G. L. A computer tool for the fusion and visualization of thermal and magnetic resonance images / G. L. Bichinho, M. A. Gariba, I. J. Sanches // 2009. — Vol. 22, No. 5. — P. 527–534.
38. Colak M. C. Predicting coronary artery disease using different artificial neural network models / M. C. Colak, H. Kocatürk, S. Sağiroğlu // 2008. — Vol. 8, No. 4. — P. 249–254.
39. Елисеева И. Общая теория статистики / И. Елисеева, М. юзбашев. — 2005. — 657 p.
40. Арлоу Д. UML 2 и унифицированный процесс. практический и объектно-ориентированный анализ и проектирование / Д. Арлоу, А. Нейштадт. — 2017. — 615 p.
41. Розенберг Д. Применение объектно- ориентированного моделирования с использованием uml и анализ прецедентов / Д. Розенберг, К. Скотт. — 2002.

42. Feldmann C. Practical guide to business process reengineering using idefo / C. Feldmann. — Addison-Wesley Professional, 2013. — 240 p.
43. Hunt B. Introduction to business process mapping with idefo & idef3 / B. Hunt. — Brian Hunt, 2013. — 19 p.
44. Cooper A. About face: the essentials of interaction design / A. Cooper, R. Reimann, D. Cronin, C. Noessel. — Indianapolis, IN : Wiley, 2014. — 720 p.
45. Beazley D. Python cookbook: recipes for mastering python 3 / D. Beazley, B. K. Jones. — O'Reilly Media, 2013. — 708 p.
46. Brett S. Effective python: 90 specific ways to write better python / S. Brett. — Addison-Wesley Professional, 2019. — 480 p.
47. Lutz M. Programming python: powerful object-oriented programming / M. Lutz. — O'Reilly Media, 2010. — 2541 p.
48. Lospinoso J. C++ crash course: a fast-paced introduction / J. Lospinoso. — No Starch Press, 2019. — 793 p.
49. Downe A. Think python: how to think like a computer scientist / A. Downe. — 2002.
50. Krug S. Don't make me think, revisited: a common sense approach to web usability / S. Krug. — Berkeley, Calif. : New Riders, 2013. — 216 p.
51. Bader D. Python tricks: a buffet of awesome python features / D. Bader. — Dan Bader, 2017. — 303 p.
52. Tidwell J. Designing interfaces: patterns for effective interaction design / J. Tidwell. — Beijing : O'Reilly Media, 2011. — 578 p.