

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти перший (бакалаврський)

**РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПОБУДОВИ МОДЕЛЕЙ**  
**ОКА, ЯКЕ ВРАЖЕНЕ ГЛАУКОМОЮ**  
(тема)

Виконав:  
студент 4 курсу, групи ІТІНФ-20-1

Антосін М.Є.  
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Керівник проф. Кузьомін О.Я.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

Кобилін О.А.  
(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)Кафедра Інформатики  
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Антосіну Микиті Євгеновичу  
(прізвище, ім'я, по батькові)1. Тема роботи Розробка інтелектуальної системи побудови моделей ока, яке вражене глаукомою

затверджена наказом університету від 20 травня 2024 року № 464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 31 травня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотеки для глибокого навчання та комп'ютерного зору TensorFlow і Keras, відкриті датасети медичних зображень, документація з використання фреймворку Flask та середовище розробки Visual Studio.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Огляд методів глибокого навчання для діагностики глаукоми на основі аналізу зображень.2. Розробка та налаштування моделей глибокого навчання з використанням TensorFlow та Keras.3. Реалізація клієнтського застосунку та мікросервісної архітектури для інтеграції моделі діагностики.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми розроблення системи діагностики глаукоми на основі глибинного навчання, постановка задачі, схема мікросервісної архітектури, графіки точності та втрат під час навчання моделі, тестові зображення фундусу ока.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.04.2024	
2	Аналіз завдання, підбір літератури	08.04.24-15.04.24	
3	Аналіз літератури з досліджуваної проблеми	16.04.24-18.04.24	
4	Аналіз технічних засобів	19.04.24-25.04.24	
5	Розробка методу	26.04.24-14.05.24	
6	Програмна реалізація	15.05.24-23.05.24	
7	Оформлення пояснювальної записки	24.05.24-26.05.24	
8	Перевірка на плагіат	27.05.24	
9	Рецензування	28.05.24	
10	Підготовка презентації та доповіді	29.05.24-02.06.24	
11	Занесення роботи в електронний архів	03.06.24	
12	Попередній захист кваліфікаційної роботи	10.06.24	

Дата видачі завдання 8 квітня 2024 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

проф. Кузьомін О.Я.  
(посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 55 с., 25 рис., 31 джерело.

МАШИННЕ НАВЧАННЯ, КОНВОЛЮЦІЙНІ НЕЙРОННІ МЕРЕЖІ, АНАЛІЗ ЗОБРАЖЕНЬ, ДІАГНОСТИКА ГЛАУКОМИ, ОБРОБКА МЕДИЧНИХ ЗОБРАЖЕНЬ.

Об'єктом роботи є аналіз зображень ока для діагностики глаукоми за допомогою машинного навчання. Метою роботи є розробка системи для моделювання ока, ураженого глаукомою, з використанням глибинного навчання та конволюційних нейронних мереж.

Використано методи конволюційних нейронних мереж для обробки зображень та автоматичної класифікації стану ока. Проведено дослідження ефективності різних архітектур у задачі розпізнавання глаукоми, та розроблено методику тренування і тестування з використанням медичних дата-сетів.

У результаті роботи створено програмний продукт для автоматичного аналізу глаукоми, що підвищує швидкість і точність діагностики порівняно з традиційними методами.

MACHINE LEARNING, CONVOLUTIONAL NEURAL NETWORKS, IMAGE ANALYSIS, GLAUCOMA DIAGNOSTICS, MEDICAL IMAGE PROCESSING.

The object of the work is the analysis of eye images for the diagnosis of glaucoma using machine learning. The aim is to develop a system for modeling eyes affected by glaucoma using deep learning methods and convolutional neural networks.

Methods of convolutional neural networks were used for image processing and automatic classification of eye condition. Research on the effectiveness of different architectures in recognizing glaucoma, and methodology for training and testing using medical datasets were conducted.

As a result, a software product for automatic glaucoma analysis was created, significantly improving the speed and accuracy of diagnostics compared to traditional methods.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	7
Вступ.....	8
1 Аналітичний огляд тематики й теорії.....	10
1.1 Машинне навчання.....	10
1.1.1 Основні поняття та методології машинного навчання.....	10
1.1.2 Google.....	11
1.1.2 OpenAI.....	13
1.2 Глаукома.....	15
1.2.1. Симптоми та ризики.....	15
1.2.2. Методи діагностики глаукоми.....	16
1.2.3 Інтеграція машинного навчання в клінічні протоколи.....	17
1.3 Постановка задачі.....	17
2 Розробка інтелектуальної системи побудови моделей.....	19
2.1 Архітектура системи.....	19
2.1.1 Мікросервісна архітектура.....	20
2.1.2 Серверлес архітектура.....	20
2.1.3 Монолітна архітектура.....	21
2.1.4 Компоненти архітектури системи.....	22
2.2 Технології.....	23
2.3 Deep learning.....	24
2.3.1 Конволюційні нейронні мережі.....	25
2.3.2 Дані для мережі.....	27
2.3.3 Аугментація даних.....	28
2.3.4 Standardized Medical Dataset for Glaucoma.....	30
2.4 Графічний інтерфейс застосунку.....	31
3 Розробка та впровадження системи.....	34
3.1 Обґрунтування вибору середовища програмної реалізації.....	34
3.2 Ініціалізація та налаштування робочого середовища.....	35
3.3 Розробка модулів системи.....	35
3.4 Кроки по запуску та виконанню програми.....	36
3.4.1 Навчання моделі.....	37
3.4.2 Розробка клієнтського застосунку.....	42
3.4.3 Розробка мікросервісної частини.....	44
3.4.4 Інструкція по запуску та використанню застосунку.....	45

3.5 Перспективи подальшої роботи.....	47
Висновки.....	50
Перелік джерел посилання.....	52

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

CNN – Convolutional neural network (конволюційні нейронні мережі)

STMG – Standardized Medical Dataset for Glaucoma

МН – машинне навчання

ШІ – штучний інтелект

ЦПУ – центральний процесор

ГПУ – графічний процесор

## ВСТУП

У сучасному світі інтелектуальні системи на основі штучного інтелекту та машинного навчання здатні суттєво покращити точність і швидкість діагностики різних захворювань, включаючи глаукому. Використання нейронних мереж та обробки зображень дозволяє автоматизувати процес діагностики, знижуючи ймовірність людської помилки та підвищуючи ефективність лікування.

Втрата зору є однією з найсерйозніших проблем, що впливають на якість життя мільйонів людей по всьому світу. Глаукома є другою найбільш поширеною причиною незворотної сліпоти, яка виникає внаслідок пошкодження зорового нерва через підвищений внутрішньоочний тиск. Своєчасна діагностика та ефективне лікування глаукоми є критично важливими для запобігання втраті зору. Проте, традиційні методи діагностики мають свої обмеження, що викликає потребу у розвитку нових технологій для покращення діагностичних можливостей.

Ця кваліфікаційна робота присвячена розробці інтелектуальної системи побудови моделей ока, які вражені глаукомою. Основною метою є створення ефективної системи для автоматичної діагностики глаукоми за допомогою глибоких нейронних мереж. У ході роботи буде використано сучасні методи машинного навчання та обробки зображень, зокрема конволюційні нейронні мережі (CNN), для аналізу зображень ока та виявлення ознак глаукоми.

Основними завданнями роботи є:

Огляд сучасних методів діагностики глаукоми та технологій, що використовуються для цього :

- підготовка та аугментація набору даних з зображеннями ока;
- розробка архітектури нейронної мережі для автоматичного виявлення глаукоми;
- навчання та тестування моделі на підготовлених даних;

– аналіз результатів та оцінка ефективності запропонованої системи.

Реалізація цієї системи дозволить суттєво покращити процес діагностики глаукоми, зробивши його більш точним та ефективним. Це, у свою чергу, сприятиме своєчасному виявленню захворювання та зменшенню кількості випадків незворотної втрати зору.

# 1 АНАЛІТИЧНИЙ ОГЛЯД ТЕМАТИКИ Й ТЕОРІЇ

## 1.1 Машинне навчання

Машинне навчання, одна з провідних технологій у світі штучного інтелекту, продовжує революціонізувати індустрію, пропонуючи все більш вдосконалені рішення для обробки великих обсягів даних. Ця технологія використовується для вирішення різноманітних завдань, починаючи від простих, таких як розпізнавання тексту або класифікація зображень, до складних, таких як розуміння різноманітних мов та автономне керування транспортними засобами [1].

### 1.1.1 Основні поняття та методології машинного навчання

Кожен день світ машинного навчання розширюється з новими поняттями та методологіями, але важливим залишається питання їх ефективності. Загальноприйнята класифікація методологій машинного навчання базується на трьох основних підходах: навчання з учителем, навчання без учителя та навчання з підкріпленням. Кожен із цих підходів має свої сильні та слабкі сторони, в залежності від конкретних вимог та контексту застосування

Навчання з учителем вважається одним із найбільш дієвих методів машинного навчання, коли маємо чітко визначені вхідні та вихідні дані. Цей метод є надзвичайно ефективним для задач регресії та класифікації. Проте, його основна вразливість полягає у великій залежності від якості та об'єму навчальних даних. Неадекватно підготовлені, упереджені чи неповні навчальні дані можуть значно знизити ефективність моделі, ведучи до неправильних передбачень на нових даних [2].

Навчання без учителя, з іншого боку, корисне для виявлення прихованих зв'язків та структур у даних без попередньо визначених міток. Цей метод ідеально підходить для задач сегментації ринку, виявлення аномалій або рекомендаційних систем. Основним недоліком є відсутність здатності до точної оцінки ефективності моделі, оскільки немає явних міток для порівняння передбачень [3].

Навчання з підкріпленням стає все більш популярним завдяки своїй здатності оптимізувати дії на основі винагород і покарань, що робить його ідеальним для розробки ігор, автоматизації роботів чи систем автоматичного водіння. Втім, великі обсяги даних для тренування, потрібні для адекватного навчання системи, та високі вимоги до обчислювальної потужності можуть стати значною перешкодою для впровадження цього методу [4].

Крім традиційних підходів, інноваційні методики, такі як глибоке навчання (deep learning), постійно розвиваються, виходячи за рамки базових алгоритмів МН.

МН стало ключовим інструментом для багатьох галузей, включаючи технологічні компанії, медицину та військову сферу. Наразі такі глобальні гіганти як Google та OpenAI активно використовують МН для покращення своїх продуктів і послуг, тоді як застосування у воєнних та медичних технологіях відкриває нові можливості та виклики.

### 1.1.2 Google

Історія Google являє перед собою довгий шлях від простої пошукової системи до світового лідера у галузі цифрових технологій та інтернет-сервісів. Він активно внедряє нові технології до своїх застосунків, та МН не стало виключенням.

Одним з видатних прикладів є пошукова система Google Search, де алгоритми МН допомагають покращити точність і релевантність пошукових результатів, швидко аналізуючи величезні обсяги інформації та визначаючи найбільш відповідні сторінки до запитів користувачів [5].

Google Translate. Цей сервіс перекладу використовує нейронні мережі, які дозволяють здійснювати більш точні і природні переклади між майже сотнею мов. Система постійно самонавчається, що дозволяє їй вдосконалюватись, аналізуючи величезні масиви тексту [6].

Google Photos впроваджує МН для розпізнавання облич, місць і предметів на зображеннях, що робить управління та пошук фотографій неймовірно зручним для користувачів. Завдяки МН, користувачі можуть шукати свої фотографії, використовуючи контентний пошук без потреби ручного тегування [7].

Область здоров'я, не стала перешкодою для Google, він допомагає мільйонам людей покращити своє здоров'я. Це – Google Health, яка застосовує МН для аналізу медичних даних та допомоги лікарям у діагностиці. Вони розробили алгоритми для виявлення діабетичної ретинопатії, аналізуючи зображення очного дна, що показує потенціал МН у покращенні медичних діагностичних систем [8].

Основною перевагою використання машинного навчання в продуктах Google є підвищення ефективності та персоналізація користувацького досвіду. Завдяки машинному навчанню Google може пропонувати користувачам більш релевантні результати пошуку, рекомендації в YouTube, персоналізовані оголошення та навіть оптимізовані маршрути в Google Maps. Ці технології дозволяють обробляти та аналізувати величезні обсяги даних у реальному часі, що значно покращує загальну продуктивність та користувацький досвід.

Проте, використання машинного навчання також створює виклики, пов'язані з конфіденційністю та безпекою даних. Оскільки алгоритми

машинного навчання потребують доступу до великих обсягів особистих даних для навчання і функціонування, питання захисту цих даних стає надзвичайно важливим. Google постійно працює над забезпеченням захисту приватності користувачів та етичним використанням машинного навчання, що є критично важливим у контексті зростаючих занепокоєнь з цих питань.

Важливо також відзначити, що Google регулярно оновлює свої політики конфіденційності та проводить аудит безпеки своїх систем, щоб виявляти та усувати потенційні загрози. Компанія інвестує значні ресурси у дослідження та розробку нових методів захисту даних, що дозволяє їй залишатися на передовій лінії боротьби з кіберзагрозами.

### 1.1.2 OpenAI

OpenAI, заснована у 2015 році, і є однією з провідних дослідницьких організацій у сфері штучного інтелекту. Її місія полягає в розробці дружнього ШІ, який буде сприяти і захищати людство в цілому. Вона відома своїми піонерськими роботами у галузі глибинного навчання та навчання з підкріпленням. Вони розробили декілька значущих і впливових моделей і систем, зокрема GPT, яка викликала широкий резонанс у галузі за свої здібності до генерації текстів, що майже не відрізняються від людських [9].

Вплив OpenAI на промисловість ШІ, розробляючи інноваційні алгоритми, які використовуються в широкому спектрі застосунків, від автоматизації текстових процесів до складних ігор, де ШІ змагається та перемагає професійних геймерів.

Наприклад, їх система Dactyl, яка навчається управляти роботичною рукою з натуралістичною декстралітетом. Як OpenAI передає ідею Dactyl – «Ми кладемо об'єкт, наприклад блок або призму, на долоню і просимо Dactyl змінити його положення; наприклад, обертаючи блок, щоб помістити нове

обличчя зверху. Мережа спостерігає лише координати кінчиків пальців і зображення з трьох звичайних RGB-камер» [9]. Вона, являється прикладом їх інноваційних робіт у фізичному світі (рис. 1.1).

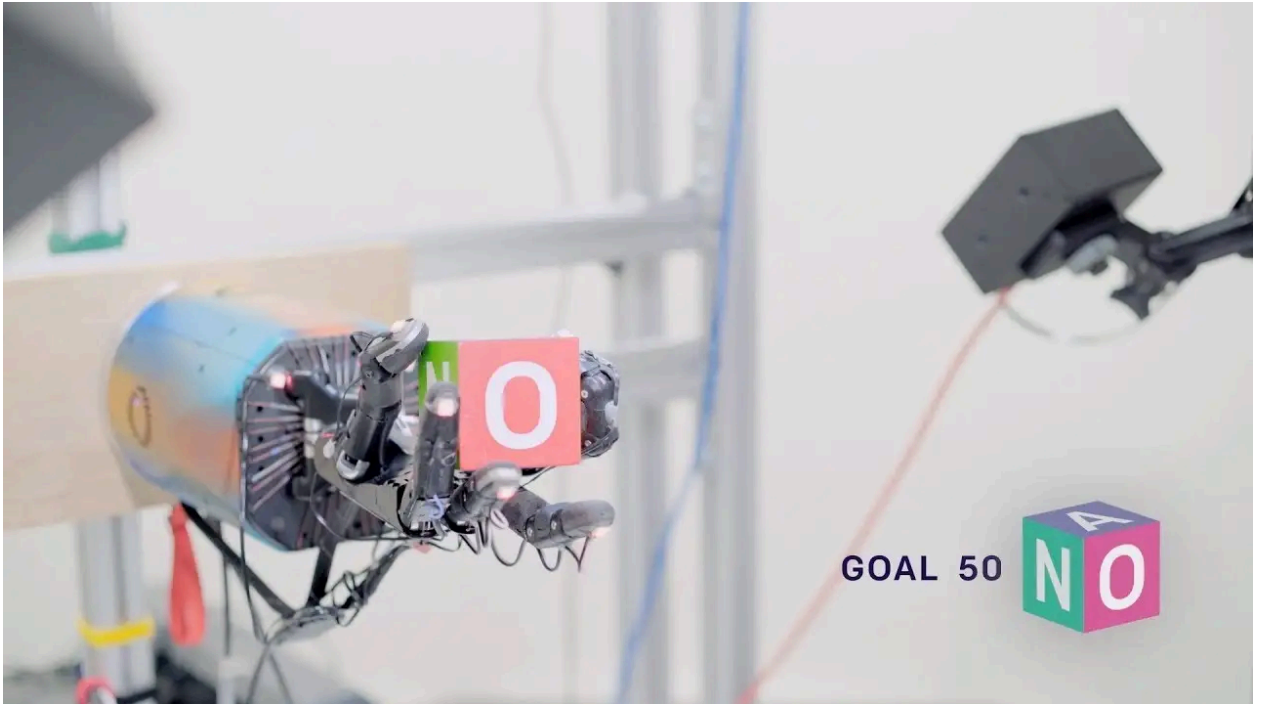


Рисунок 1.1 – Система Dactyl

Одним з ключових досягнень OpenAI є створення GPT-4, однієї з найпотужніших моделей для генерації тексту, здатної генерувати зв'язні та змістовні текстові відповіді на запити, що мають велике значення для розвитку ботів-помічників, освітніх програм та інших інтерактивних застосунків. GPT-4 здатен не тільки відповідати на запитання, але й створювати вражаючі літературні твори, сценарії, технічну документацію та багато іншого. Ця модель вже використовується в численних застосунках для автоматизації рутинних завдань, підтримки користувачів та навчання.

OpenAI не зупиняється на досягнутому і планує подальший розвиток своїх моделей. У майбутньому очікується анонс нових версій GPT, які зможуть аналізувати людські емоції та надавати ще більш персоналізовані відповіді. Це відкриє нові можливості для створення інноваційних сервісів та

продуктів, що значно підвищують якість взаємодії користувачів з технологіями штучного інтелекту.

Такі вдосконалення дозволяють використовувати GPT-4 не лише як інструмент для автоматизації та підтримки, але й для глибокого розуміння людських потреб та настроїв, що зробить взаємодію з технологіями більш природною та ефективною.

## 1.2 Глаукома

Глаукома – це група офтальмологічних захворювань, які характеризуються пошкодженням зорового нерва, що часто пов'язане з підвищеним тиском всередині ока. Це захворювання є однією з провідних причин необоротної сліпоти у світі. Оскільки симптоми часто не виявляються до того, як почнеться значна втрата зору, рання діагностика та лікування мають критичне значення.

Згідно зі статистикою, глаукома впливає на близько 60 мільйонів осіб у світі, з яких приблизно 8 мільйонів страждають від повної сліпоти. Частота захворюваності значно зростає з віком, особливо після 60 років, але глаукома може вражати осіб будь-якого віку, включаючи новонароджених. В Україні захворюваність на глаукому також залишається високою, що вимагає покращення методів діагностики та лікування [10].

### 1.2.1. Симптоми та ризики

Глаукома часто називається «тихим злодієм зору», тому що більшість форм цього захворювання прогресує без помітних симптомів до того, як зір починає погіршуватися. Вона зазвичай діагностується під час стандартного

офтальмологічного обстеження, де можуть бути виявлені підвищений внутрішньоочний тиск і зміни у зоровому нерві. Втрата зору зазвичай починається з периферійного зору, поступово переходячи до центрального, що може призвести до повної сліпоти.

### 1.2.2. Методи діагностики глаукоми

Щоб не стати жертвою «тихого злодія зору», треба щороку проходити обстеження. Існує багато тестів, завдяки яким, можна вчасно почати процедури лікування, а саме :

- тонометрія – це тест для вимірювання внутрішньоочного тиску, за допомогою приладу, званого тонометром. Підвищений тиск у оці є основним фактором ризику для розвитку глаукоми, хоча глаукома може розвиватися і при нормальному тиску [11];

- периметрія (тест поля зору) – допомагає виявити ранні зміни в периферійному зорі, які можуть вказувати на розвиток глаукоми [11];

- гоніоскопія – цей тест дозволяє оцінити кут передньої камери ока. Закриття цього кута може призвести до різкого підвищення внутрішньоочного тиску і є типовим для гострої закритокутової глаукоми [11];

- офтальмоскопія – це дослідження дозволяє лікарю оглянути зоровий нерв на наявність змін, типових для глаукоми [11].

З появою технологій машинного навчання і штучного інтелекту, з'явилися нові можливості для покращення діагностики глаукоми. Машинне навчання може бути використане для аналізу медичних зображень з високою точністю.

Застосування нейронних мереж для аналізу зображень фундусу ока дозволяє автоматично визначати ознаки глаукоми, включаючи зміни у зоровому нерві та інші аномалії, які можуть вказувати на глаукому. Ці

системи здатні обробляти велику кількість даних швидше та з більшою точністю, ніж людські офтальмологи.

### 1.2.3 Інтеграція машинного навчання в клінічні протоколи

Інтеграція машинного навчання в клінічні протоколи відкриває нові можливості для покращення якості медичних послуг, оптимізації лікування та персоналізації підходів до пацієнтів. Це особливо актуально у галузях, де потрібен високий рівень точності діагностики і велика кількість даних для аналізу, як от в офтальмології для діагностики глаукоми [12].

## 1.3 Постановка задачі

Об'єктом роботи є аналіз зображень ока для діагностики глаукоми за допомогою машинного навчання. Метою роботи є розробка системи для моделювання ока, ураженого глаукомою, з використанням глибинного навчання та конволюційних нейронних мереж.

Для досягнення мети необхідно вирішити такі завдання:

- аналіз наукових робіт та методів: провести глибокий аналіз існуючих наукових досліджень та методів діагностики глаукоми, з особливим акцентом на застосуванні методів машинного навчання та аналізу зображень очного дна;
- розробка алгоритмів на базі CNN: розробити алгоритми на основі конволюційних нейронних мереж, здатні автоматично виявляти і класифікувати ознаки глаукоми на зображеннях очного дна. Ці алгоритми повинні бути адаптовані для різних стадій захворювання та різних типів даних зображень [13];

- тестування та валідація моделі: провести тестування розроблених алгоритмів з використанням стандартних наборів даних (datasets) для оцінки їх ефективності, точності та здатності узагальнювати. Валідація моделі здійснюватиметься через співпрацю з медичними фахівцями;
- клінічна інтеграція: Розгортання та інтеграція системи в клінічні установи для практичного використання, що включає в себе розробку інтерфейсу користувача, який буде зручним і інтуїтивно зрозумілим для медичних працівників;
- створення користувацького інтерфейсу: Розробити та налаштувати графічний інтерфейс користувача, який дозволить лікарям ефективно взаємодіяти з системою, забезпечуючи легкий доступ до результатів аналізу, історії пацієнта та інших необхідних функцій.

## 2 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПОБУДОВИ МОДЕЛЕЙ

### 2.1 Архітектура системи

Побудова архітектури системи включає в себе ряд заходів, що допомагають чітко визначити, як саме буде розроблена система. В даному випадку ми розглядаємо верхній рівень визначення архітектури, що можна визначити при обмеженій кількості інформації. Важливо зрозуміти, що не існує універсального рішення при визначенні принципу проєктування архітектури. Неможливо сказати, що конкретний архітектурний патерн підходить для будь-якої задачі і його слід використовувати виключно. У кожному випадку все залежить від конкретного об'єкту та його завдань [14].

Хоча можна побудувати будь-яку систему на будь-якій архітектурі, це не завжди є доцільним рішенням. Загалом виділяють кілька типів верхньорівневої архітектури застосунків [14]:

- мікросервісна архітектура;
- серверлес архітектура;
- монолітна архітектура.

Для розробки інтелектуальної системи для діагностики глаукоми на зображеннях ока, різні архітектури програмного забезпечення можуть вплинути на ефективність, швидкість розробки, а також на легкість майбутнього масштабування та підтримки.

Вибір архітектури для системи діагностики глаукоми залежить від конкретних вимог об'єкту, включаючи обсяг даних, що обробляються, необхідний рівень продуктивності та надійності, а також ресурси, доступні для розробки та підтримки системи.

### 2.1.1 Мікросервісна архітектура

Мікросервіси дозволяють розбити застосунок на незалежні сервіси, кожен з яких виконує певну функцію та комунікує з іншими через визначені API. Кожен сервіс може бути розроблений, розгорнутий, масштабований та відновлений незалежно від інших [16].

Переваги :

- гнучкість та масштабування: окремі компоненти можуть бути масштабовані в залежності від потреб;
- спрощення управління: відмова одного компонента не впливає на роботу інших;
- технологічна незалежність: можливість вибору найкращих технологій для кожного сервісу окремо.

Недоліки :

- складність управління: керування численними сервісами може бути складнішим, ніж управління одним монолітним застосунком;
- складність розробки: розробники повинні враховувати взаємодію між сервісами, що може збільшити складність коду;
- витрати на інфраструктуру: через необхідність окремого розгортання сервісів можуть зрости загальні витрати на інфраструктуру.

### 2.1.2 Серверлес архітектура

Серверлес-архітектура передбачає використання облачних сервісів, які автоматично управляють алокацією ресурсів. Розробники фокусуються на написанні коду без необхідності керування серверами [17].

Переваги :

- зниження вартості: оплата лише за використані обчислювальні ресурси;
- швидкість розгортання: легке додавання нових функцій та сервісів;
- автоматичне масштабування: ресурси автоматично нарощуються та зменшуються в залежності від навантаження.

Недоліки :

- обмежені можливості контролю: зменшення контролю над серверним середовищем та операційними можливостями;
- залежність від постачальника: тісний зв'язок з облачними сервісами може призвести до «vendor lock-in» ефекту, що ускладнює міграцію на інші платформи;
- холодний старт: затримки при ініціалізації нових інстансів, які можуть негативно вплинути на продуктивність.

### 2.1.3 Монолітна архітектура

Монолітний застосунок – це єдиний незалежний блок, у якому всі компоненти застосунку тісно інтегровані та залежні один від одного [17].

Переваги :

- простота розробки та розгортання: легше управляти одним застосунком, ніж кількома;
- легкість тестування: всі компоненти працюють у єдиному середовищі.

Недоліки :

- важке масштабування: всі компоненти застосунку масштабуються разом, що може бути неефективним та дорогим;

- складність внесення змін: оновлення одного компонента може вимагати перезапуску всього застосунку;
- високий ризик збою: помилка в одному компоненті може призвести до збою всього застосунку.

#### 2.1.4 Компоненти архітектури системи

Для інтелектуальної системи діагностики глаукоми було обрано мікросервісну архітектуру, оскільки вона забезпечує гнучкість, масштабованість і високу доступність, що є критично важливим для аналізу зображень та взаємодії з користувачем у реальному часі (рис. 2.1).

Інтелектуальної система діагностики глаукоми складається з кількох основних компонентів :

- клієнтський застосунок: інтерфейс користувача, де користувач завантажує зображення та отримує результати аналізу;
- серверна частина: компонент, який отримує запити від клієнтського додатку, обробляє їх та повертає результати;
- модель глибокого навчання: нейронна мережа, яка здійснює аналіз зображень для виявлення ознак глаукоми;
- база даних: компонент для зберігання даних, таких як історія завантажених зображень та результати аналізу.

Правильний вибір архітектури дозволяє досягти високої продуктивності, масштабованості та надійності системи. Враховуючи вимоги до обробки зображень у реальному часі та високої точності діагностики, було обрано архітектуру на основі мікросервісів, яка дозволяє розділити функціональність на окремі компоненти, кожен з яких відповідає за певну задачу.

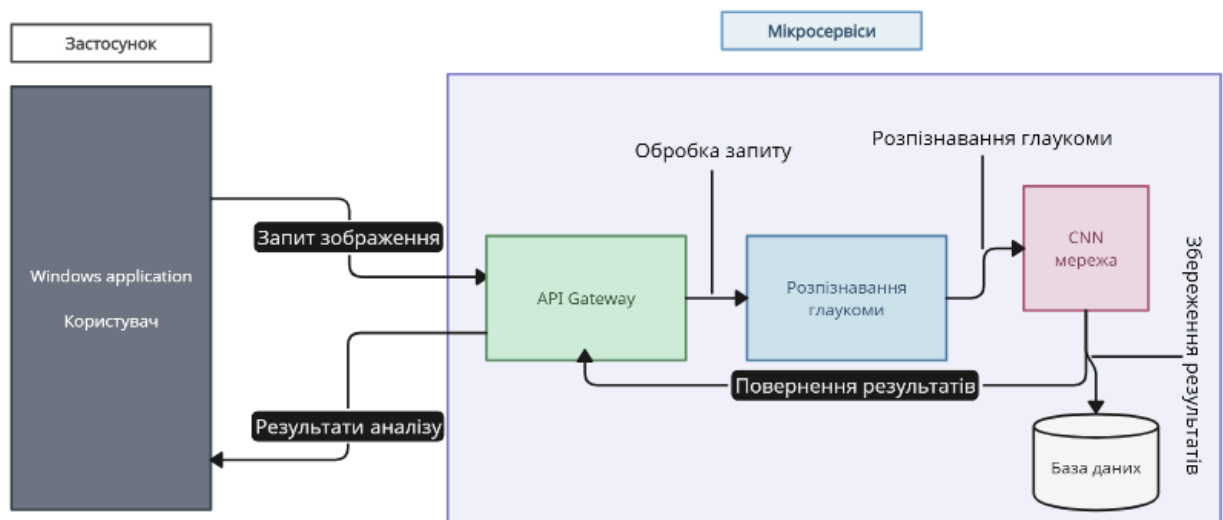


Рисунок 2.1 – Архітектура застосунку

## 2.2 Технології

Вибір правильних технологій є фундаментальним аспектом у розробці будь-якого програмного застосунку, оскільки це визначає не тільки можливості розширення та ефективності застосунку, але й впливає на швидкість його розробки та підтримки. Для проекту інтелектуальної системи діагностики глаукоми було обрано низку технологій, які дозволяють ефективно вирішувати задачі поставлені перед системою, а саме :

- ImageDataGenerator – це клас у Keras, який використовується для генерації тензорів зображень у реальному часі з аугментацією даних. Це допомагає розширити розмір вашого набору даних та покращити узагальнювальну здатність моделі. Дозволяє здійснювати такі трансформації зображень, як горизонтальне відображення, обертання, масштабування та зміна яскравості, що робить модель більш стійкою до різноманітних варіацій вхідних даних [19-21];

– Flask – це легкий веб-фреймворк для Python, який використовується для створення веб-застосунків. Він забезпечує зручний спосіб створення RESTful API для взаємодії з моделлю глибокого навчання. Використовується для розгортання моделі на сервері та створення веб-інтерфейсу для завантаження зображень та отримання результатів аналізу. Flask забезпечує швидку і зручну інтеграцію з моделями TensorFlow та Keras [19-21];

– Matplotlib – це бібліотека для створення статичних, анімаційних та інтерактивних візуалізацій на мові Python. Використовується для візуалізації результатів навчання моделі. Дозволяє створювати графіки точності та втрат, що допомагає аналізувати ефективність навчання моделі та виявляти можливості для покращення [22];

– C# – це мова програмування, яка широко використовується для розробки додатків на платформі .NET. Windows Forms — це бібліотека для створення графічного інтерфейсу користувача на Windows. C# та Windows Forms використовуються для створення десктопного застосунку, який дозволяє користувачам легко взаємодіяти із системою діагностики глаукоми. Графічний інтерфейс користувача дозволяє завантажувати зображення, відправляти їх на сервер для аналізу та отримувати результати [23].

### 2.3 Deep learning

Глибоке навчання є основним інструментом сучасного машинного навчання, який імітує роботу біологічних нейронних мереж, що складають мозок (рис. 2.2). Вони складаються з великої кількості простих обчислювальних одиниць, які називаються нейронами, з'єднаних між собою зваженими зв'язками. Основна ідея нейронних мереж полягає в тому, що

вони можуть навчатися виконувати завдання шляхом налаштування ваг зв'язків на основі прикладів навчання [13].

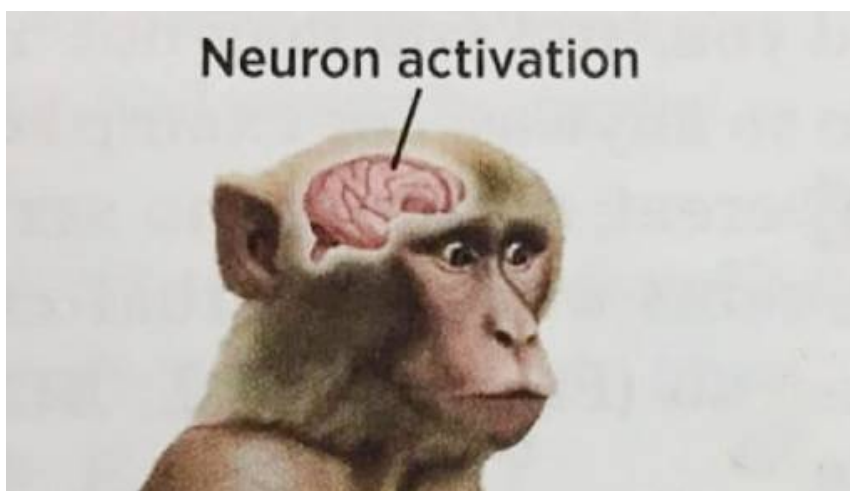


Рисунок 2.2 – Нейронне навчання

### 2.3.1 Конволюційні нейронні мережі

Для задачі діагностики глаукоми було обрано використання конволюційної нейронної мережі. CNN є спеціалізованим типом нейронної мережі, який особливо добре підходить для обробки та класифікації зображень. Основна перевага CNN полягає в її здатності автоматично виділяти важливі ознаки з зображень за допомогою конволюційних шарів [13].

CNN складаються з кількох шарів, які автоматично виділяють ознаки зображень різного рівня абстракції [13]. Використання CNN дозволяє досягти високої точності виявлення глаукоми завдяки здатності мереж навчатися на великій кількості даних і виявляти складні патерни, які можуть бути невидимими для людського ока (рис.2. 3).

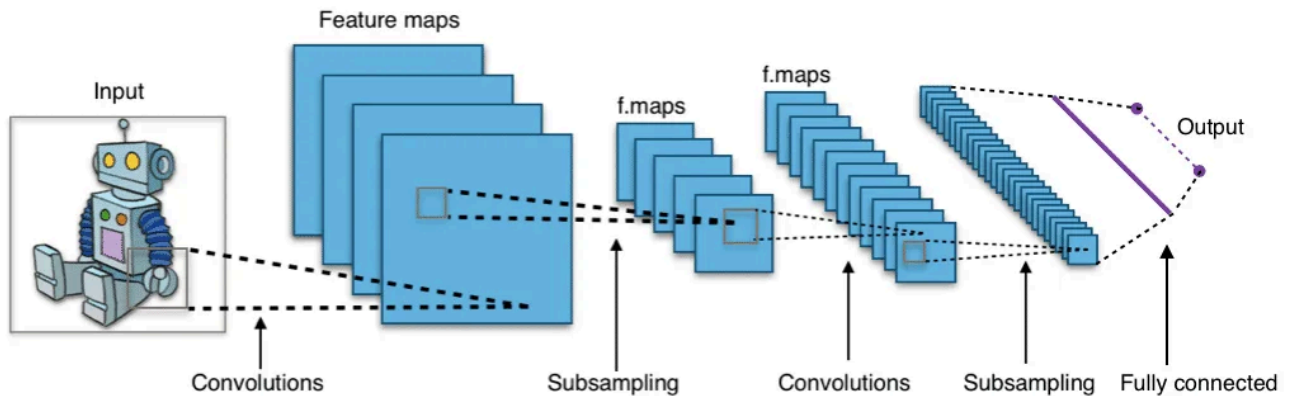


Рисунок 2.3 – CNN мережі

Конволюційні шари є основними будівельними блоками CNN. Вони використовують фільтри для обробки вхідних зображень і виділення ознак, таких як контури, текстури та інші деталі. Кожен фільтр застосовується до всього зображення, створюючи карту ознак, яка потім передається до наступного шару [19].

Шари підвибірки (MaxPooling) зменшують розмірність карт ознак, що дозволяє зменшити кількість параметрів у моделі та покращити її узагальнювальну здатність. MaxPooling вибирає максимальне значення з кожного підвибіркового вікна, що зберігає найбільш важливі ознаки та зменшує розмір даних [19].

Шари нормалізації (BatchNormalization) стабілізують та прискорюють навчання моделі, нормалізуючи виходи попередніх шарів. Це зменшує вплив зміни параметрів на наступні шари та допомагає уникнути перенавчання [19].

Повнозв'язні шари (Dense) об'єднують всі ознаки, виділені попередніми шарами, та використовуються для остаточної класифікації. Кожен нейрон у повнозв'язному шарі з'єднаний з усіма нейронами попереднього шару, що дозволяє моделі приймати рішення на основі всіх доступних ознак [19].

Архітектура нейронної мережі для діагностики глаукоми зазвичай включає кілька конволюційних шарів, за якими слідують шари підвибірки (MaxPooling) та повнозв'язні шари (Dense) [19]. Додатково можуть використовуватися шари нормалізації (BatchNormalization) та регуляризації (Dropout), які допомагають запобігти перенавчанню моделі (рис.2.4) [19].

```
# Створення моделі
import tensorflow as tf
model = tf.Sequential([
    layers.Dense(64, activation='relu', input_shape=(input_dim,)),
    layers.Dense(32, activation='relu'),
    layers.Dense(1, activation='sigmoid')# Один нейрон у вихідному шарі для бінарної класифікації
])
```

Рисунок 2.4 – Приклад ініціалізації CNN моделі

Переваги використання CNN для виявлення глаукоми :

- Автоматичне виділення ознак: CNN автоматично виділяє важливі ознаки з зображень, що робить їх ідеальними для задач розпізнавання зображень;
- Масштабованість: CNN можуть обробляти великі набори даних та ефективно використовувати обчислювальні ресурси, такі як GPU;
- Висока точність: CNN демонструють високу точність у задачах класифікації зображень та виявлення патологій.

### 2.3.2 Дані для мережі

Дата-сети відіграють ключову роль у процесі навчання машинного навчання та глибокого навчання, зокрема у створенні ефективних і точних моделей для різноманітних задач, включаючи діагностику глаукоми за допомогою зображень ока. Це особливо важливо у медичній інформатиці, де

якість і різноманітність даних можуть значно вплинути на висновки, до яких приходять моделі, та її здатність узагальнювати нові дані поза навчальним набором.

Дата-сет – це колекція даних, яка зазвичай використовується для навчання машинного навчання або алгоритмів глибокого навчання [24]. Дата-сети для медичних зображень, які використовуються для діагностики глаукоми, містять тисячі зображень фундуса, які лікарі або спеціалізовані медичні працівники анотували, вказавши наявність або відсутність глаукоми. Їх якість безпосередньо впливає на ефективність навчання моделі. Чим вища різноманітність і точність анотацій у дата-сеті, тим краще модель може навчитися розпізнавати складні медичні зображення та узагальнювати свої висновки на нових даних. Недоліки в дата-сеті, такі як помилки в анотаціях, недостатня репрезентативність зразків різних демографічних груп, можуть призвести до упередженості моделі та помилкових діагнозів. При роботі з медичними зображеннями виникають специфічні виклики. Перше — це забезпечення конфіденційності та безпеки пацієнтської інформації. Кожен дата-сет має бути анонімізований і відповідати медичним нормам і стандартам, таким як HIPAA в США або GDPR у Європі [25]. Інший виклик — це необхідність мінімізації упередженості моделі, що вимагає акуратної балансування та представлення різних демографічних груп у тренувальних даних.

### 2.3.3 Аугментація даних

Одним із способів покращення здатності моделі до узагальнення є аугментація даних – процес генерації нових тренувальних даних з існуючих

шляхом застосування різноманітних трансформацій, таких як обертання, зсуви, зміна масштабу та яскравості.



Рисунок 2.5 – Аугментація даних

Це допомагає моделі стати менш чутливою до варіацій у вхідних даних, що зустрічаються в реальному світі, і підвищує її точність при роботі з новими зображеннями [19] (рис. 2.5).

Основні методи аугментації даних [19]:

- `rescale`: масштабування значень пікселів зображень до діапазону  $[0, 1]$ . Це допомагає нормалізувати дані та прискорює процес навчання;
- `horizontal flip`: горизонтальне відображення зображень. Ця трансформація збільшує кількість зразків і допомагає моделі стати стійкішою до варіацій у даних;
- `rotation range`: обертання зображень на випадковий кут у заданому діапазоні (наприклад, від  $-20$  до  $20$  градусів). Це дозволяє моделі навчитися розпізнавати ознаки незалежно від орієнтації зображень.

- zoom range: масштабування зображень на випадкове значення в заданому діапазоні. Це допомагає моделі стати стійкішою до зміни масштабу об'єктів на зображеннях;
- width shift range: зміщення зображень по горизонталі на випадкову величину в заданому діапазоні. Це допомагає моделі стати стійкішою до зміщень об'єктів на зображеннях;
- height shift range: зміщення зображень по вертикалі на випадкову величину в заданому діапазоні. Це також допомагає моделі стати стійкішою до зміщень об'єктів на зображеннях;
- brightness range: зміна яскравості зображень у заданому діапазоні. Це дозволяє моделі стати стійкішою до змін умов освітлення на зображеннях.

#### 2.3.4 Standardized Medical Dataset for Glaucoma

Для розробки інтелектуальної системи побудови моделей ока, які вражені глаукомою, був обраний дата-сет SMDG, що розшифровується як Standardized Medical Dataset for Glaucoma, є одним із визначальних дата-сетів у дослідженнях, пов'язаних з глаукомою [26]. Цей дата-сет створено з метою сприяння розвитку і удосконалення алгоритмів машинного навчання та комп'ютерного зору, які можуть автоматично діагностувати і оцінювати стадії глаукоми на основі зображень сітківки. Він включає понад 12000 зображень фундусу ока, які зібрані з різних клінічних джерел (рис.2.6). Кожне зображення в цьому дата-сеті анотоване медичними фахівцями, що надає точні відомості про стан сітківки і наявність або відсутність глаукоми. Це робить SMDG особливо цінним для розробників діагностичного програмного забезпечення, оскільки забезпечує високу точність та надійність даних для тренування та тестування їхніх моделей [26].

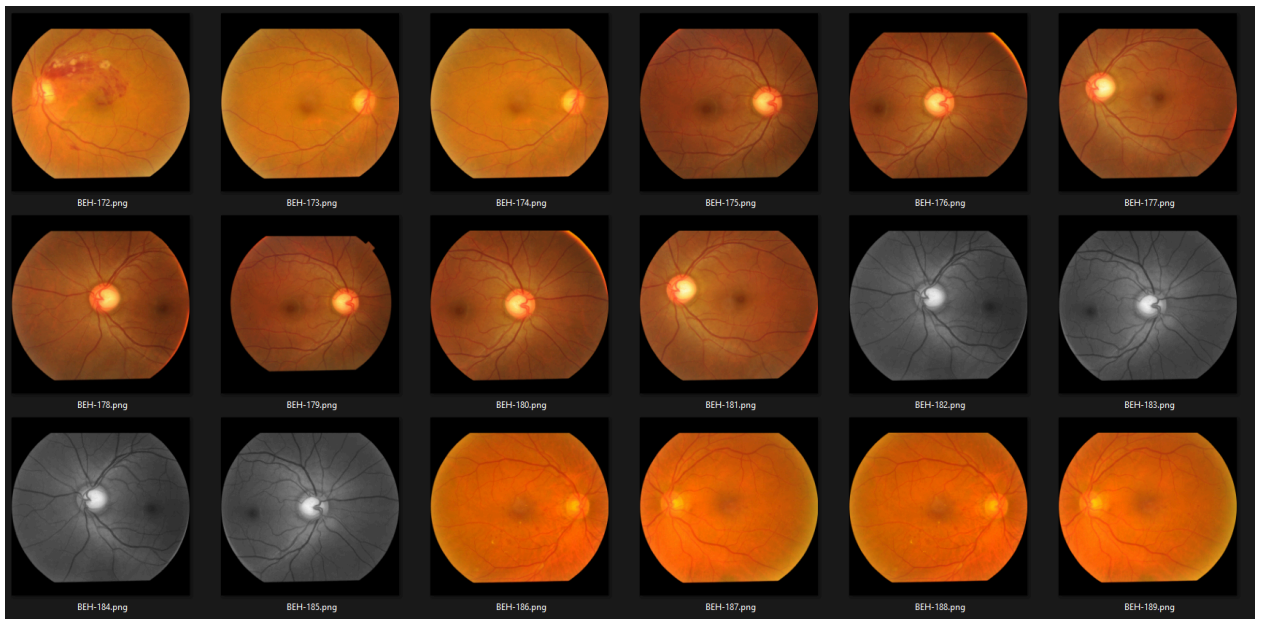


Рисунок 2.6 – SMDG дата-сет

SSMDG, розроблений за підтримки численних медичних установ і наукових організацій, є важливим ресурсом у дослідженні захворювань на глаукому. Цей дата-сет широко доступний для наукової спільноти через відомі онлайн платформи для обміну даними, такі як Kaggle, де він був опублікований для загального доступу [26].

## 2.4 Графічний інтерфейс застосунку

Графічний інтерфейс користувача (Graphical User Interface, GUI) є ключовим компонентом будь-якого програмного забезпечення, що забезпечує зручність взаємодії користувача із застосунком. Основне завдання GUI — зробити програму інтуїтивно зрозумілою та легкою у використанні, забезпечуючи доступ до всіх необхідних функцій та інформації.

Графічний інтерфейс складається з візуальних елементів, таких як вікна, кнопки, меню, поля вводу та інші елементи управління, які дозволяють користувачам взаємодіяти з програмою. На відміну від текстового інтерфейсу, GUI використовує візуальні метафори, що робить роботу з програмою більш інтуїтивною.

Основні компоненти графічного інтерфейсу:

- вікна: основні контейнери для відображення інформації та елементів управління;
- кнопки: елементи для виконання дій або команд;
- поля вводу: елементи для введення тексту або числових даних;
- меню: списки команд та опцій для вибору;
- графічні елементи: іконки, зображення та графіки для візуалізації даних та інформації.

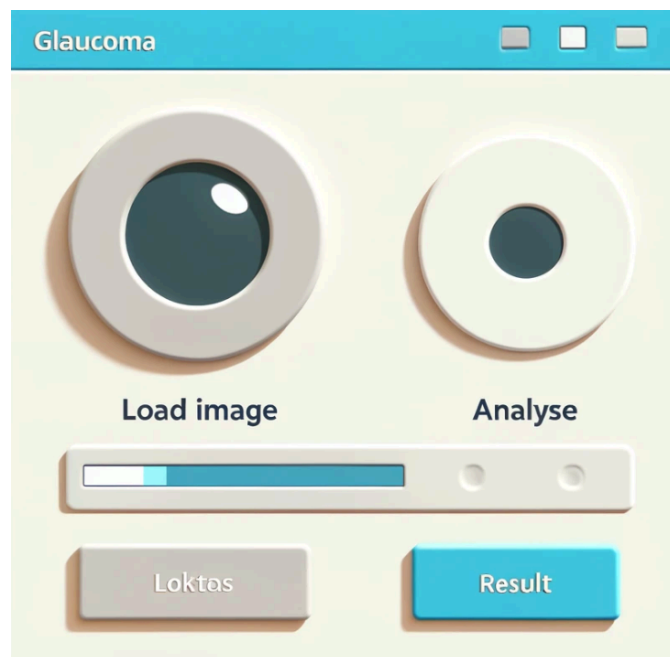


Рисунок 2.7 – Макет інтерфейсу користувача

Під час розробки графічного інтерфейсу для застосунку важливо враховувати такі аспекти:

- інтуїтивність: інтерфейс повинен бути зрозумілим та легким у використанні без необхідності навчання;
- ергономічність: всі елементи повинні бути розташовані логічно та зручно для користувача;
- візуальна привабливість: дизайн інтерфейсу повинен бути привабливим та сучасним, сприяючи позитивному враженню від використання застосунку;
- функціональність: інтерфейс повинен забезпечувати доступ до всіх необхідних функцій та інформації без зайвих кроків;
- адаптивність: інтерфейс повинен добре виглядати та функціонувати на різних пристроях з різними розмірами екрану.

## 3 РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ

### 3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи була розроблена CNN мережа для визначення результатів дослідження ока, яке вражено глаукомою, графічний застосунок та мікросервіс на базі Rest API. Для реалізації було обрано комбінацію Python та бібліотек TensorFlow і Keras для роботи з нейронними мережами, Flask для розробки серверної частини та C# з Windows Forms для створення настільного додатку, що слугуватиме як клієнтський інтерфейс.

Обрання Python та його екосистеми для машинного навчання обумовлено широким співтовариством, великою кількістю готових рішень та бібліотек, які спрощують процес розробки та скорочують час впровадження нових функціональних можливостей. TensorFlow та Keras дозволяють ефективно створювати, тренувати та імплементувати глибокі нейронні мережі, що є ключем до вирішення задач класифікації зображень на предмет виявлення глаукоми.

Використання Flask як легкого вебфреймворку дозволяє швидко створювати та тестувати API для взаємодії між сервером і клієнтом. Flask підтримує легку інтеграцію з Python-орієнтованими бібліотеками машинного навчання і володіє достатньою гнучкістю для розгортання рішень в інтернеті.

З іншого боку, використання C# та Windows Forms для розробки настільної частини дозволяє створити стабільний та функціональний інтерфейс користувача. Ці інструменти відмінно підходять для швидкої розробки додатків з багатим графічним інтерфейсом, які легко впроваджувати і тестувати в середовищах Windows.

### 3.2 Ініціалізація та налаштування робочого середовища

Перед початком розробки системи було налаштовано робоче середовище, що включає інсталяцію необхідних бібліотек і засобів розробки. Встановлення Python, налаштування інтегрованого середовища розробки (IDE) та залучення необхідних залежностей, таких як TensorFlow, Keras, і Flask, було здійснено за допомогою системи управління пакетами pip.

Для серверної частини було встановлено Flask разом із необхідними розширеннями для роботи з HTTP запитами і відповідями. Клієнтська частина, написана на C#, була розроблена у середовищі Visual Studio, що забезпечує міцні інструменти для дизайну інтерфейсу та дебагінгу програмного коду.

### 3.3 Розробка модулів системи

Процес розробки інтелектуальної системи діагностики глаукоми можна розділити на кілька ключових модулів:

- модуль глибокого навчання для аналізу зображень;
- серверний модуль для обробки запитів та відповідей;
- клієнтський застосунок для взаємодії з користувачами.

Кожен з цих модулів був ретельно спланований та розроблений для забезпечення ефективної взаємодії та обміну даними між компонентами системи. Тестування інтеграції компонентів та їх взаємодія зі зберігачем даних було ключовим для забезпечення стабільності та надійності роботи системи.

### 3.4 Кроки по запуску та виконанню програми

Для забезпечення належного середовища для навчання моделі була обрана версія Python 3.7, оскільки це остання підтримувана версія на Windows для запуску навчання як на ЦПУ, так і на ГПУ.

Процес включає наступні кроки: встановлення Python 3.7 з офіційного сайту (рис. 3.1), налаштування інтегрованого середовища розробки (IDE) та встановлення необхідних бібліотек, таких як TensorFlow, Keras та Flask за допомогою системи управління пакетами pip (рис. 3.2).

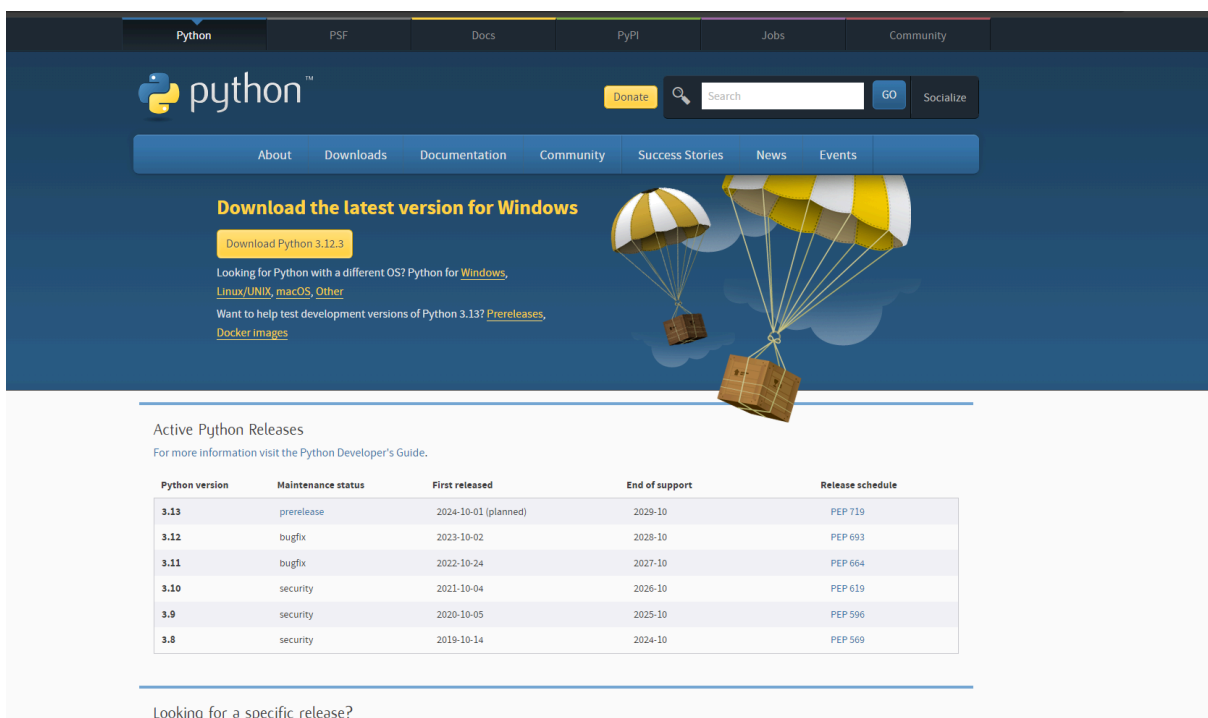


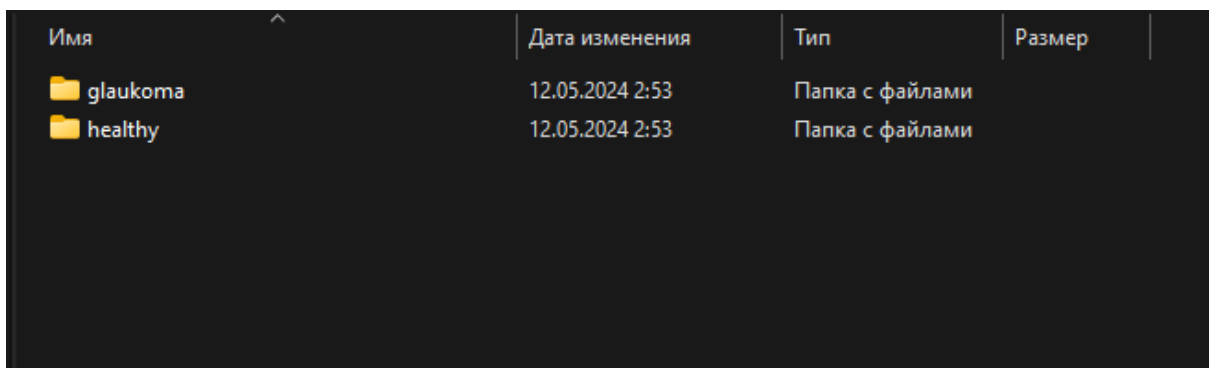
Рисунок 3.1 – Встановлення Python

```
PowerShell 7.4.2
PS C:\Users\Nikita> pip install tensorflow keras flask pillow
```

Рисунок 3.2 – Встановлення Tensorflow

### 3.4.1 Навчання моделі

Крок 1. Для навчання моделі використовувався дата-сет SMDG (Standardized Medical Dataset for Glaucoma), який згадується у другому розділі. Цей дата-сет був завантажений в локальну середу для навчання нейронної мережі. Він містить тисячі зображень фундусу ока, які анотовані медичними фахівцями і містять інформацію про наявність або відсутність глаукоми (рис.3.3).



Имя	Дата изменения	Тип	Размер
glaucoma	12.05.2024 2:53	Папка с файлами	
healthy	12.05.2024 2:53	Папка с файлами	

Рисунок 3.3 – Директорія зі знімками

Крок 2 . За допомогою бібліотеки tensorflow проводимо підготовку даних (рис. 3.4).

Крок 3 . Ініціалізація моделі (рис. 3.5) , початок навчання, (рис. 3.6). Архітектура нейронної мережі, яка використовується для діагностики глаукоми (рис. 3.7). Щоб зрозуміти якість навчання було додано візуалізацію результатів навчання (рис. 3.8).

Крок 5. Після тривалої тренування, яка тривала приблизно 10 годин, була отримана модель нейронної мережі та графік її навчання. Цей графік демонструє зміни точності та втрат на тренувальних і валідаційних даних протягом епох навчання (рис. 3.9).

```

48 trainDataGen = ImageDataGenerator(preprocessing_function=preprocess_input)
49
50 trainGen = trainDataGen.flow_from_dataframe(
51     dataframe=train,
52     batch_size=batchSize,
53     shuffle=True,
54     x_col="combined",
55     y_col="types",
56     validate_filenames=True,
57     target_size=(target, target),
58     directory='ДАТАСЕТЬ/6L_different_types/full-fundus/full-fundus/',
59     color_mode='rgb'
60 )
61
62 testDataGen = ImageDataGenerator(preprocessing_function=preprocess_input)
63
64 testGen = testDataGen.flow_from_dataframe(
65     dataframe=test,
66     batch_size=1,
67     shuffle=False,
68     x_col="combined",
69     y_col="types",
70     validate_filenames=True,
71     target_size=(target, target),
72     directory='ДАТАСЕТЬ/6L_different_types/full-fundus/full-fundus/',
73     color_mode='rgb'
74 )
75

```

Рисунок 3.4 – Код для підготовки даних

```

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(512, 512, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')
])

```

Рисунок 3.5 – Ініціалізація нейронної мережі

```

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

history = model.fit(
    trainGen,
    steps_per_epoch=len(trainGen),
    validation_data=testGen,
    validation_steps=len(testGen),
    epochs=10
)

```

Рисунок 3.6 – Запуск тренування нейронної мережі

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 510, 510, 32)	896
max_pooling2d (MaxPooling2D)	(None, 255, 255, 32)	0
batch_normalization (BatchNormalization)	(None, 255, 255, 32)	128
conv2d_1 (Conv2D)	(None, 253, 253, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 126, 126, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 126, 126, 64)	256
conv2d_2 (Conv2D)	(None, 124, 124, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 62, 62, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 62, 62, 128)	512
flatten (Flatten)	(None, 492032)	0
dense (Dense)	(None, 128)	62,980,224
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129

Рисунок 3.7 – Архітектура нейронної мережі

```

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

losss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(epochs)

plt.figure(figsize=(8, 8))
plt.subplot(*args: 1, 2, 1)
plt.plot(*args: epochs_range, acc, label='Training Accuracy')
plt.plot(*args: epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(*args: 1, 2, 2)
plt.plot(*args: epochs_range, loss, label='Training Loss')
plt.plot(*args: epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()

```

Рисунок 3.8 – Візуалізація результатів навчання



Рисунок 3.9 – Графік точності та втрат під час навчання

Процес навчання включав кілька епох, кожна з яких містила багато ітерацій через весь набір даних. Протягом цих епох модель поступово навчалася розпізнавати патерни, характерні для знімків ока, уражених глаукомою (рис. 3.10).

```

1245/1245 ————— 1468s 1s/step - accuracy: 0.6940 - loss: 0.6622 - val_accuracy: 0.3905 - val_loss: 1.8643
Epoch 2/10
1245/1245 ————— 1541s 1s/step - accuracy: 0.7821 - loss: 0.4829 - val_accuracy: 0.4476 - val_loss: 1.4437
Epoch 3/10
1245/1245 ————— 1453s 1s/step - accuracy: 0.8137 - loss: 0.4381 - val_accuracy: 0.4194 - val_loss: 1.7585
Epoch 4/10
1245/1245 ————— 1476s 1s/step - accuracy: 0.8148 - loss: 0.4224 - val_accuracy: 0.3982 - val_loss: 2.2680
Epoch 5/10
1245/1245 ————— 1454s 1s/step - accuracy: 0.8453 - loss: 0.3772 - val_accuracy: 0.4198 - val_loss: 2.4547
Epoch 6/10
1245/1245 ————— 1452s 1s/step - accuracy: 0.8691 - loss: 0.3453 - val_accuracy: 0.3973 - val_loss: 4.3562
Epoch 7/10
1245/1245 ————— 1451s 1s/step - accuracy: 0.8655 - loss: 0.3384 - val_accuracy: 0.3805 - val_loss: 3.4184
Epoch 8/10
1245/1245 ————— 1449s 1s/step - accuracy: 0.8728 - loss: 0.3345 - val_accuracy: 0.4010 - val_loss: 3.9759
Epoch 9/10
1245/1245 ————— 1451s 1s/step - accuracy: 0.8735 - loss: 0.3224 - val_accuracy: 0.3849 - val_loss: 7.5420
Epoch 10/10
1245/1245 ————— 1453s 1s/step - accuracy: 0.8798 - loss: 0.3287 - val_accuracy: 0.4042 - val_loss: 4.6868
312/312 ————— 53s 169ms/step - accuracy: 0.4149 - loss: 4.7140
Validation Accuracy: 40.42%

```

Рисунок 3.10 – Графік точності та втрат під час навчання

### Крок 5. Аналіз результатів навчання

На графіках «Training and Validation Accuracy» та «Training and Validation Loss» можна побачити наступні тенденції (рис 3.9):

- графік точності (accuracy): навчальна точність постійно зростає і досягає близько 0.9, що свідчить про успішне навчання моделі. Водночас валідаційна точність залишається значно нижчою, коливаючись близько 0.4-0.5, що може вказувати на перенавчання моделі (overfitting).

- графік втрат (loss): навчальні втрати постійно знижуються, що підтверджує зростання точності на навчальній вибірці. Однак валідаційні втрати мають нестабільну поведінку з значними коливаннями та високими значеннями, що також підтверджує проблему перенавчання.

Натренерована модель виявилася не такою точною, як очікувалося. Точність за оцінками тестування досягає лише 40,42%, але на практиці вона значно нижча. Це вказує на те, що модель не здатна адекватно узагальнювати на нових даних, і потрібно провести додаткові дослідження та оптимізацію для поліпшення її продуктивності.

### 3.4.2 Розробка клієнтського застосунку

Крок 1. Про'єктування інтерфейсу користувача. Інтерфейс складається з декількох ключових компонентів, таких як кнопки для завантаження зображення та його аналізу, індикаторів процесу та полів для виводу результатів. (рис. 3.11).

```

btnAnalyse.BackColor = Color.Transparent;
btnAnalyse.FlatStyle = FlatStyle.Flat;
btnAnalyse.FlatAppearance.BorderSize = 0;
btnAnalyse.Paint += (sender, object?, e : PaintEventArgs) => PaintRoundButton(sender, e, Color.Gray);
btnAnalyse.Click += new EventHandler(this.BtnAnalyse_Click);

// Load Image label
lblLoadImage = new Label();
lblLoadImage.Text = "Заргузити знімок";
lblLoadImage.Location = new Point(x: 120, y: 250);
lblLoadImage.Size = new Size(width: 140, height: 40);
lblLoadImage.TextAlign = ContentAlignment.MiddleCenter;

// Analyse label
lblAnalyse = new Label();
lblAnalyse.Text = "Аналізувати";
lblAnalyse.Location = new Point(x: 500, y: 250);
lblAnalyse.Size = new Size(width: 140, height: 40);
lblAnalyse.TextAlign = ContentAlignment.MiddleCenter;

// Loading Icon
loadingIcon = new PictureBox();
loadingIcon.Size = new Size(width: 128, height: 128);
loadingIcon.Location = new Point(x: 330, y: 280);
loadingIcon.SizeMode = PictureBoxSizeMode.StretchImage;
loadingIcon.Image = Image.FromFile($"{AppDomain.CurrentDomain.BaseDirectory}/Pic/Loading.gif");
loadingIcon.Visible = false;

// Image button
btnImage = new Button();
btnImage.Size = new Size(width: 160, height: 60);
btnImage.Location = new Point(x: 120, y: 400);
btnImage.Text = "Побачити знімок";
btnImage.BackColor = Color.FromArgb(red: 240, green: 240, blue: 240);
btnImage.FlatStyle = FlatStyle.Flat;
btnImage.FlatAppearance.BorderSize = 0;
btnImage.Click += new EventHandler(this.LblLoadImage_Click);

```

Рисунок 3.11 – Код інтерфейсу користувача клієнтського застосунку

Крок 2. Для забезпечення можливості завантаження зображень у програму, було створено функцію для обробки події натискання на кнопку «Завантажити знімок». Ця функція відкриває діалогове вікно для вибору файлу, завантажує обране зображення і відображає його у відповідному полі інтерфейсу (рис. 3.12).

```
private void LbLoadImage_Click(object sender, EventArgs e)
{
    if (loadedImage != null)
    {
        Form popupForm = new Form();
        popupForm.Text = "Знімок";
        popupForm.Size = new Size((int)loadedImage.Width + 20, (int)loadedImage.Height + 40);
        popupForm.FormBorderStyle = FormBorderStyle.FixedDialog;
        popupForm.MaximizeBox = false;
        popupForm.MinimizeBox = false;

        PictureBox pictureBox = new PictureBox();
        pictureBox.Image = loadedImage;
        pictureBox.SizeMode = PictureBoxSizeMode.StretchImage;
        pictureBox.Dock = DockStyle.Fill;

        popupForm.Controls.Add(pictureBox);
        popupForm.ShowDialog();
    }
    else
    {
        MessageBox.Show("Будь-яке обрано не правильний тип файлу", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Рисунок 3.12 – Код для завантаження зображення

Крок 3. Наступним було додавання можливості аналізу завантаженого зображення за допомогою CNN мережі. Для цього була створена функція для обробки події натискання на кнопку «Аналізувати». Вона відправляє зображення на сервер для аналізу через REST API, отримує результат і відображає його користувачеві (рис. 3.13).

```
private async void btnAnalyse_Click(object sender, EventArgs e)
{
    btnAnalyse.Paint += (sender, g) => PaintRoundButton(sender, e, Color.Gray);
    if (loadedImage != null)
    {
        try
        {
            loadingIcon.Visible = true;
            bytes[] imageBytes;
            using (var ms = new System.IO.MemoryStream())
            {
                loadedImage.Save(ms, loadedImage.RawFormat);
                imageBytes = ms.ToArray();
            }

            using (HttpClient client = new HttpClient())
            {
                using (var content = new MultipartFormDataContent())
                {
                    content.Add(new ByteArrayContent(imageBytes), "file", filename: "image.jpg");
                    HttpResponseMessage response = await client.PostAsync("http://127.0.0.1:5000/check", content);
                    response.EnsureSuccessStatusCode();
                    string responseBody = await response.Content.ReadAsStringAsync();
                    var color = JsonConvert.DeserializeObject<Color>(responseBody);
                    string text;
                    if (color == Color.Red)
                    {
                        btnAnalyse.Paint += (sender, g) => PaintRoundButton(sender, e, Color.Red);
                        text = "помилка";
                    }
                    else
                    {
                        btnAnalyse.Paint += (sender, g) => PaintRoundButton(sender, e, Color.GreenYellow);
                        MessageBox.Show("Analysis result: " + responseBody, "Analysis Result", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error during analysis: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        finally
        {
            loadingIcon.Visible = false;
        }
    }
    else
    {
        MessageBox.Show("No image loaded. Please load an image first.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Рисунок 3.13 – Код для відправки зображення на сервер і отримання результату

### 3.4.3 Розробка мікросервісної частини

Для забезпечення взаємодії між клієнтським застосунком і сервером було реалізовано REST API, яке дозволяє завантажувати зображення, отримувати результати аналізу та керувати іншими функціями системи. Це API було розроблено за допомогою Flask та відповідних розширень (рис. 3.14).

```
1  from flask import Flask, request, jsonify
2  import tensorflow as tf
3  import io
4  from PIL import Image
5  import numpy as np
6
7  app = Flask(__name__)
8
9  # Load the model
10 model = tf.keras.models.load_model('glaucoma_detection_model.h5')
11
12 @app.route(rule='/check', methods=['POST'])
13 def upload_file():
14     if request.method == 'POST':
15         file = request.files['file']
16         if file:
17
18             img_bytes = file.read()
19             img = Image.open(io.BytesIO(img_bytes))
20             img = img.resize((512, 512))
21             img_array = tf.keras.preprocessing.image.img_to_array(img)
22             img_array = tf.expand_dims(img_array, 0)
23
24             # Get the prediction
25             prediction = model.predict(img_array)
26             result = bool(prediction[0][0] > 0.5)
27
28             return jsonify({'glaucoma': result})
29         return jsonify({'Помилка': 'Не має файлу'}), 400
30
31 if __name__ == "__main__":
32     app.run(debug=True)
```

Рисунок 3.14 – Код реалізації REST API

Мікросервіс був інтегрований з попередньо навченою моделлю нейронної мережі, що дозволяє йому проводити аналіз зображень у режимі реального часу.

### 3.4.4 Інструкція по запуску та використанню застосунку

Загальні поради та рекомендації:

- перед завантаженням зображення переконайтеся, що воно має достатньо високу якість та чіткість. Розпізнавання зображень буде кращим на якісних зображеннях;
- уникайте завантаження занадто великих файлів зображень, оскільки це може сповільнити процес аналізу та обробки;
- перевірте, чи зображення відповідає вимогам до формату (JPEG, PNG тощо).

Крок 1. Вхід до застосунку. Відкрийте застосунок, двічі клацнувши на його значок або запустивши його через меню «Пуск».

Крок 2. Завантаження зображення. Натисніть на кнопку «Завантажити знімок» (рис. 3.15).

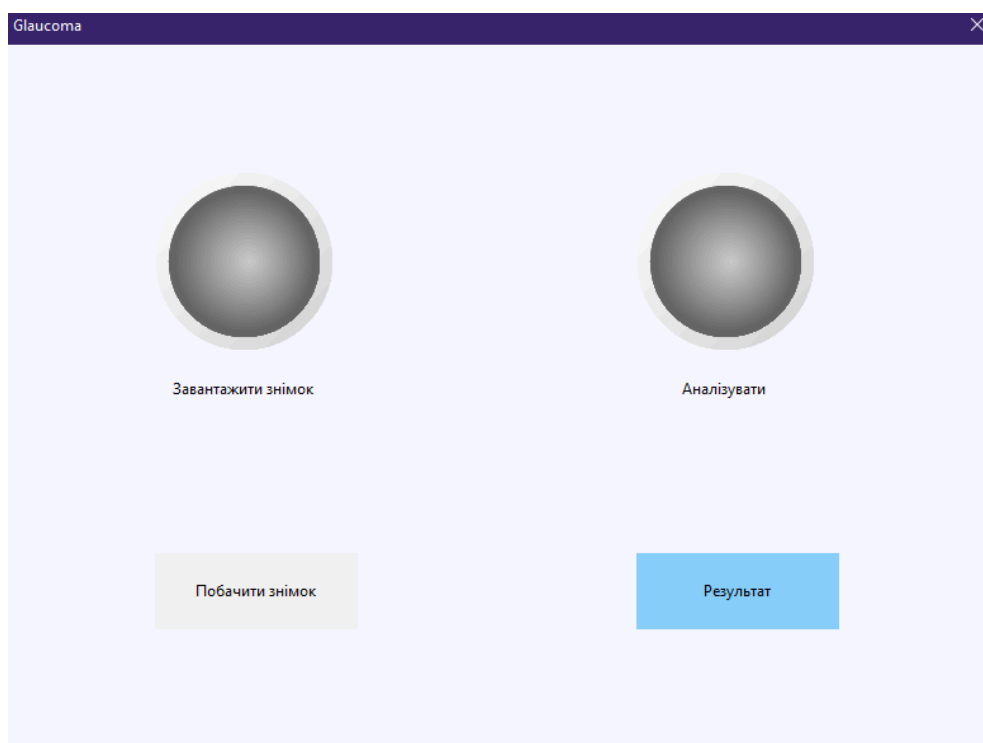


Рисунок 3.15 – Інтерфейс застосунку

У відкритому діалоговому вікні виберіть зображення з вашої локальної файлової системи. Після вибору необхідного файлу натисніть «Відкрити» або аналогічну кнопку у діалоговому вікні (рис. 3.16). Завантажене зображення буде додано до застосунку, де його можна переглянути, натиснувши кнопку «Переглянути знімок».



Рисунок 3.16 – Обрати знімок для аналізу

Крок 3. Аналіз зображення. Натисніть на кнопку «Аналізувати» (рис. 3.17). Застосунок відправить зображення на сервер для аналізу через REST API. Під час цього процесу ви побачите іконку завантаження, яка буде обертатися, вказуючи на те, що процес аналізу триває.

Після завершення аналізу результат буде відображено у відповідному полі інтерфейсу.

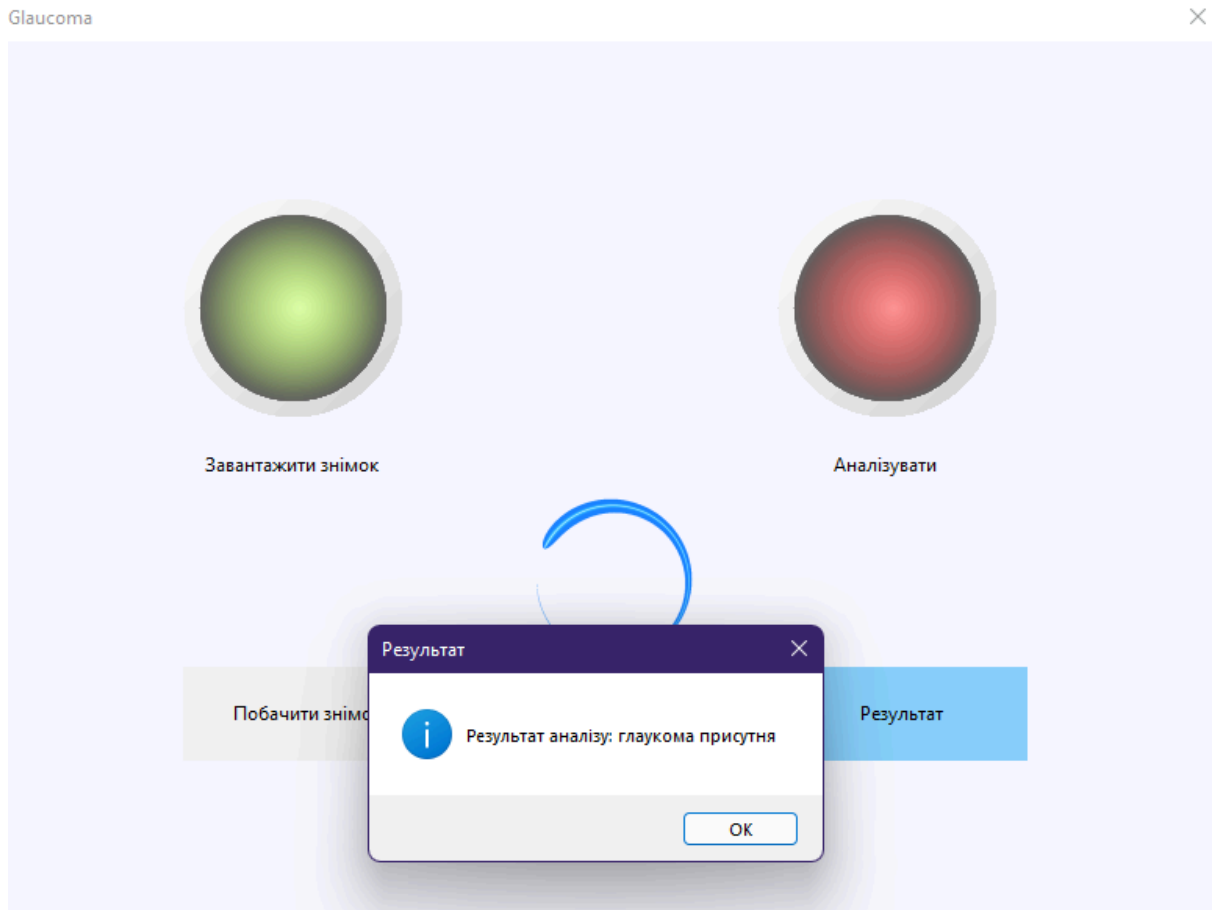


Рисунок 3.17 – Результат аналізу

#### Крок 4. Завершення роботи

Закрийте застосунок, натиснувши на кнопку «Вихід» або просто закривши вікно застосунку.

### 3.5 Перспективи подальшої роботи

Під час подальшої роботи планується зосередитися на вдосконаленні точності моделі для діагностики глаукоми. Основними напрямками покращень будуть:

- оптимізація архітектури нейронної мережі: проведення досліджень та експериментів з різними архітектурами CNN для досягнення

кращої узагальнюючої здатності та зниження проблеми перенавчання. Це включатиме вивчення сучасних підходів, таких як ResNet, Inception, EfficientNet та інших передових моделей;

- аугментація даних: використання більш широкого набору методів аугментації даних для підвищення стійкості моделі до різних варіацій зображень. Це може включати обертання, зміщення, зміну яскравості та контрасту, додавання шуму та інші методи, що допоможуть моделі краще адаптуватися до реальних умов;

- збір та інтеграція додаткових даних: розширення тренувального дата-сету шляхом додавання нових зображень фонду оку з різних джерел. Це дозволить покращити різноманітність даних та забезпечити модель більш репрезентативними зображеннями для навчання;

- регуляризація моделі: впровадження технік регуляризації, таких як Dropout, Batch Normalization, L2-регуляризація та інших методів для зменшення перенавчання та підвищення стабільності моделі;

- покращення інтерфейсу користувача: зосередження на покращенні інтерфейсу користувача для більш зручного використання клієнтського застосунку. Це може включати перегляд та оптимізацію розташування елементів, удосконалення графічного дизайну, додавання нових функцій для поліпшення користувацького досвіду;

- додавання нових можливостей: додавання нових функцій до системи, які розширяють її функціональність та роблять її більш корисною. Це можуть бути додаткові опції для аналізу зображень, інтеграція з іншими медичними системами, можливість зберігання та перегляду історії аналізів та інші корисні функції;

- забезпечення надійності та безпеки: приділення уваги безпеці та стабільності системи. Важливо забезпечити, щоб нові функції були добре інтегровані в існуючу систему, працювали стабільно та безпечно. Будуть

проводитися тестування та збір фідбеку від користувачів для виявлення можливих проблем та вдосконалення функціоналу згідно з їх потребами.

Загалом, подальша робота над системою спрямована на забезпечення найкращого користувальницького досвіду і задоволення потреб користувачів. Покращення точності моделі, інтерфейсу користувача та додавання нових можливостей дозволять розширити функціонал системи та забезпечити більш широкий спектр використання.

## ВИСНОВКИ

У рамках кваліфікаційної роботи була розроблена та реалізована інтелектуальна система для діагностики глаукоми, що включає глибоку нейронну мережу (CNN), графічний застосунок та мікросервіс на базі REST API.

За допомогою сучасних методів глибинного навчання та бібліотек TensorFlow і Keras була створена модель нейронної мережі для аналізу зображень фундусу ока. Основна мета проєкту полягала в забезпеченні високої точності діагностики глаукоми, тому для досягнення цього були використані потужні алгоритми глибинного навчання та ефективні методи обробки зображень.

Система була піддана ретельному аналізу та тестуванню з використанням великого набору зображень фундусу ока, анотованих медичними фахівцями. Процес тестування включав оцінку точності моделі на нових даних, що дозволило виявити проблеми перенавчання та провести відповідну оптимізацію. У результаті було досягнуто певного покращення точності моделі, хоча залишаються виклики, пов'язані з її здатністю узагальнювати на нових даних.

Графічний застосунок, розроблений на основі C# та Windows Forms, забезпечує зручний інтерфейс для користувачів, що дозволяє легко завантажувати зображення, відправляти їх на аналіз та отримувати результати діагностики. Веб-фреймворк Flask був використаний для реалізації серверної частини, що забезпечує обробку запитів та взаємодію з моделлю нейронної мережі.

Проведені дослідження також включали аналіз можливостей подальшого вдосконалення системи з метою покращення її функціоналу та користувацького досвіду. Було виявлено, що для підвищення точності

діагностики необхідно розширити дата-сет, застосовувати додаткові методи аугментації даних та оптимізувати архітектуру нейронної мережі.

Результати роботи мають значення для розвитку систем діагностики глаукоми за допомогою методів машинного навчання у медичних закладах. Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «СУЧАСНІ АСПЕКТИ ТА ПЕРСПЕКТИВНІ НАПРЯМКИ РОЗВИТКУ НАУКИ»

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Machine learning URL: [https://en.wikipedia.org/wiki/Machine\\_learning](https://en.wikipedia.org/wiki/Machine_learning) (дата звернення 19.05.2024).
2. Supervised learning URL: [https://en.wikipedia.org/wiki/Supervised\\_learning](https://en.wikipedia.org/wiki/Supervised_learning) (дата звернення 25.04.2024).
3. Unsupervised learning URL: [https://en.wikipedia.org/wiki/Unsupervised\\_learning](https://en.wikipedia.org/wiki/Unsupervised_learning) (дата звернення 13.05.2024).
4. Reinforcement learning URL: [https://en.wikipedia.org/wiki/Reinforcement\\_learning](https://en.wikipedia.org/wiki/Reinforcement_learning) (дата звернення 15.04.2024).
5. Google search URL: [https://en.wikipedia.org/wiki/Google\\_Search](https://en.wikipedia.org/wiki/Google_Search) (дата звернення 14.05.2024).
6. Google translate URL: [https://en.wikipedia.org/wiki/Google\\_Translate](https://en.wikipedia.org/wiki/Google_Translate) (дата звернення 08.05.2024).
7. Google photos URL: [https://en.wikipedia.org/wiki/Google\\_Photos](https://en.wikipedia.org/wiki/Google_Photos) (дата звернення 10.05.2024).
8. Google health URL: [https://en.wikipedia.org/wiki/Google\\_Health](https://en.wikipedia.org/wiki/Google_Health) (дата звернення 03.05.2024).
9. OpenAI or open-source artificial intelligence URL: <https://en.wikipedia.org/wiki/OpenAI> (дата звернення 20.05.2024).
10. Glaucoma URL: <https://en.wikipedia.org/wiki/Glaucoma> (дата звернення 19.04.2024).
11. Методи діагностики глаукоми. URL: <https://www.visiobud.com/uk/diagnostika-glaukomy-v-kieve/> (дата звернення 19.04.2024).
12. Sandhu, S., Lin, A. L., Brajer, N., Sperling, J., Ratliff, W., Bedoya, A. D., ... & Sendak, M. P. (2020). Integrating a machine learning system into clinical workflows: qualitative study. *Journal of Medical Internet Research*, 22(11), e22421.

13. Oguz, C., Aydin, T., & Yaganoglu, M. (2024). A CNN-based hybrid model to detect glaucoma disease. *Multimedia Tools and Applications*, 83(6), 17921-17939.
14. Software architecture. URL: [https://en.wikipedia.org/wiki/Software\\_architecture](https://en.wikipedia.org/wiki/Software_architecture) (дата звернення 14.05.2024).
15. Kuzomin O. \_ Creation of intelligent systems for analyzing supermarket visitors to identify criminal elements. Kuzomin O. \_ I. \_ , Berkovskyi D. \_ S. \_ // May 5, 2023 | Sydney, Australia | Collection of scientific papers &quot;SCIENTIA&quot;. SECTION 13. INFORMATION TECHNOLOGIES AND SYSTEMS. pp. 113-118
16. Microservices. URL : <https://en.wikipedia.org/wiki/Microservices> (дата звернення 14.05.2024)
17. Болож, О. (2023). Огляд serverless архітектури та її переваги. Матеріали VI Міжнародної студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання“, 110-111.
18. Monolithic Architecture або монолітна архітектура. URL : <https://iampm.club/ua/blog/shho-krashhe-monolit-chi-mikroservisi-yak-obrati-arhitekturu-projektu/> (дата звернення 14.05.2024)
19. Kuzomin O. Development and research of image segmentation using mask r- cnn , grabcut and opencv , International scientific journal &quot;Grail of Science&quot;/ Terebetskyi MA, Kuzomin OY 2022, 14/15, pp. 362-368.
20. Kuzomin O. Investigation and investigation segmentation picture for help mask r - cnn , grabcut and opencv / Terebetsky M., Kuzomin O.// Grail of Science. 2022. No. 14-15. pp. 362–368.
21. Kuzomin O. Rozrobka that investigation detector masks For exposing opencv , keras / tensorflow and deep \_ navchannyam . G. S. Verkolab , O. Ya . Kuzyomin . Technologies and strategies for the implementation of scientific achievements: collection of additional materials \_ &quot;SCIENTIA&quot; I

Mizhnar . sc. - theory. conf. T. 2, herbs. 27, 2022. Stockholm, SWE: European Scientific Platform. pp. 78–83 .

22. Matplotlib: Visualization with Python. URL : <https://matplotlib.org/> (дата звернення 15.05.2024)

23. C# language documentation. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення 13.05.2024)

24. Quiñero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (Eds.). (2022). Dataset shift in machine learning. Mit Press.

25. Phillips, M. (2018). International data-sharing norms: from the OECD to the General Data Protection Regulation (GDPR). Human genetics, 137, 575-582.

26. SMDG, A Standardized Fundus Glaucoma Dataset. URL : <https://www.kaggle.com/datasets/deathtrooper/multichannel-glaucoma-benchmark-dataset> (дата звернення 18.05.2024)

27. Kuzomin O. Creation of intelligent systems for analyzing supermarket visitors to identify criminal elements / Berkovskyi D., Kuzomin O. Collection of Scientific Papers “SCIENTIA”, (May 5, 2023; Sydney, Australia), 113–118.

28. Kuzomin O. Ya . Modeling of repetitions to understand in unbalanced data flows / SE Kholodov , Kuzomin O. Ya . // Development of the scientific thought of post-industrial marriage: current discourse: materials of the III International Scientific Conference, April 28, 2023, Lviv . - Vinnytsia : European Science Platform, 2023. - P. 112-119.

29. O. Kuzomin ABOUT. I. \_ Follow-up methods contextually - stale recognition of human activities behind I'll help you hybrid models deep navchannya / M. D'yachenko , O. Kuzomin // Theoretically that practical zastosuvannya results daily sciences: materials of the V International student scientific conference, June 27, 2023. – M. Rivne, Ukraine - p. 121-124.

30. Kuzomin O. Rozvitok scientific thoughts of the post-industrial marriage: current discourse 106 Section XV. information technology and systems

gnuchka virobnycha system based on python from keruvannyam for multi-agent support navchannya Koidan A. A . . Kuzomin O. May 5, 2023 | Sydney, Australia  
Collection of scientific papers “SCIENTIA”;

31. Kuzomin O. Methods analysis depict legen For diagnosis of COVID / VO Prokipets , O. Ya .Kuzyomin // International scientific journalgrail of science / VO Prokipets , O. Ya . Kuzyomin . - Viden , 2022. pp. 356–361.