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 АНАЛІЗУ ЗОБРАЖЕНЬ ДЛЯ ВИЯВЛЕННЯ МІТОК
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-2
Софіч Р. О.
(прізвище, ініціали)

Спеціальність 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

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Софічу Руслану Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка аналізу зображень для виявлення міток

затверджена наказом університету від 19 травня 2025 року № 381Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 7 червня 2025 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі.

4. Перелік питань, що потрібно опрацювати в роботі
1. Огляд методів детектування елементів на зображеннях сітківки ока (оптичний диск, виїмка) з акцентом на згорткові нейронні мережі.
2. Побудова та аналіз математичної моделі сегментації структур очного дна (на прикладі U- Net).
3. Розрахунок CDR для виявлення глаукоми.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми, постановка задачі, тестові зображення.

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	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів	15.04.25-20.04.25	
5	Розробка методу	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

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

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 77 с., 2 табл., 8 рис., 47 джерел.

ГЛАУКОМА, ОПТИЧНИЙ ДИСК, ГЛИБИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, U-NET, CDR, СЕГМЕНТАЦІЯ, КЛАСИФІКАЦІЯ.

Об'єктом роботи є зображення очного дна, зокрема структура зорового нерва.

Метою даної роботи є створення програмного засобу для автоматизованого аналізу зображень сітківки ока з використанням методів глибинного навчання, що дозволить здійснювати ефективну діагностику глаукоми та оцінювати ризик її розвитку.

У межах роботи побудовано нейронну модель U-Net для сегментації оптичного диска й виїмки, а також класифікатор для виявлення глаукоми. Для підвищення точності застосовано трансферне навчання, аугментацію зображень та нормалізацію даних.

Розроблено два варіанти програмного забезпечення: десктопний застосунок на основі PyQt та вебінтерфейс на Flask + React для інтерактивного аналізу. В основі системи — датасет DRISHTI-GS з анотаціями. Реалізовано обчислення CDR та оцінку ефективності моделі за метриками точності, чутливості, специфічності, AUC та IoU.

GLAUCOMA, OPTIC DISC, DEEP LEARNING, COMPUTER VISION, U-NET, CDR, SEGMENTATION, CLASSIFICATION.

The object of this work is fundus images, particularly the structure of the optic nerve.

The aim of this work is to create a software tool for automated analysis of retinal images using deep learning methods, which will allow for effective diagnosis of glaucoma and assessment of the risk of its development.

Within the project, a U-Net neural network model was developed for segmenting the optic disc and cup, along with a classifier for glaucoma detection. To improve accuracy, transfer learning, image augmentation, and data normalization were applied.

Two software versions were implemented: a desktop application based on PyQt and a web interface using Flask + React for interactive analysis. The system uses the DRISHTI-GS dataset with annotations. It performs CDR calculation and evaluates model performance with accuracy, sensitivity, specificity, AUC, and IoU metrics.

ЗМІСТ

Вступ 6
1 Аналіз предметної області та постановка задачі 10
1.1 Огляд проблеми діагностики захворювань сітівки ока 10
1.2 Аналіз методів комп’ютерної обробки зображень сітківки 12
1.3 Огляд існуючих рішень та алгоритмів для виявлення глаукоми ... 15
1.4 Формування вимог та постановка задачі дослідження 18
2 Математичне обґрунтування вибраних методів 22
2.1 Теоретичні основи згорткових нейронних мереж для аналізу медичних зображень 22
2.2 Архітектура U-Net для сегментації зображень сітківки 27
2.3 Методи виявлення діагностичних міток глаукоми 31
2.4 Оцінка ефективності та метрики сегментації зображень 37
3 Реалізація постановки задачі та тестування 44
3.1 Архітектура програмного додатку 44
3.2 Підготовка та обробка даних 46
3.3 Процес навчання нейронної мережі 53
3.4 Графічний інтерфейс користувача 58
3.5 Тестування та аналіз результатів роботи системи 63
Висновки 69
Перелік джерел посилання 71

ВСТУП

Захворювання сітківки ока, зокрема глаукома [1], є однією з провідних причин незворотної втрати зору в усьому світі. Згідно з даними Всесвітньої організації охорони здоров’я, понад 76 мільйонів людей у світі страждають від глаукоми, при цьому приблизно 60% випадків залишаються невиявленими на ранніх стадіях через безсимптомний перебіг захворювання. Рання діагностика глаукоми має вирішальне значення для запобігання незворотній втраті зору, оскільки лікування, розпочате на початкових стадіях, може суттєво сповільнити прогресування захворювання та зберегти зорові функції пацієнта.

Традиційна діагностика захворювань сітківки ґрунтується на візуальному аналізі зображень очного дна (фундус-фотографії) та оптичної когерентної томографії (ОСТ [4]) офтальмологами. Цей процес має суттєві обмеження, зокрема трудомісткість, суб’єктивність інтерпретації та нерівномірний доступ до спеціалізованої офтальмологічної допомоги. Офтальмолог повинен детально аналізувати знімки, що вимагає значних часових витрат та високої кваліфікації, а в умовах зростаючого навантаження на медичні заклади це призводить до обмеження кількості пацієнтів, яких може обстежити спеціаліст.

Проблема нерівномірного доступу до спеціалізованої офтальмологічної допомоги особливо гостро стоїть у сільській місцевості та в країнах з низьким та середнім рівнем доходу, де спостерігається дефіцит кваліфікованих офтальмологів. Це призводить до затримки діагностики та початку лікування, що критично для захворювань, ефективність терапії яких значною мірою залежить від ранньої діагностики. Варіабельність діагностичних висновків між різними спеціалістами також створює додаткові виклики для стандартизації оцінки та моніторингу прогресування захворювання.

Автоматизація процесу аналізу зображень сітківки з використанням методів комп’ютерного зору та штучного інтелекту представляє

перспективний напрямок для вирішення зазначених проблем. Сучасні технології глибинного навчання, зокрема згорткові нейронні мережі, продемонстрували значний потенціал у автоматичній діагностиці захворювань сітківки з точністю, порівнянною або навіть вищою за точність [36] діагностики досвідченими офтальмологами. Такі системи дозволяють не лише прискорити процес аналізу, але й забезпечити його стандартизацію та об’єктивність.

Розробка автоматизованих систем для діагностики глаукоми та інших захворювань сітківки має значний соціально-економічний ефект. З одного боку, вона дозволяє розширити доступ до скринінгу для широких верств населення, включаючи віддалені регіони через системи телемедицини. З іншого боку, рання діагностика [3] та своєчасне лікування глаукоми зменшують витрати на медичну допомогу, пов’язану з пізніми стадіями захворювання та інвалідністю через втрату зору. Таким чином, створення ефективних систем автоматизованої діагностики глаукоми на основі аналізу зображень сітківки є актуальною науково-практичною задачею, вирішення якої має значний вплив на якість офтальмологічної допомоги та збереження зору мільйонів людей у всьому світі.

Метою даної роботи є створення програмного засобу для автоматизованого аналізу зображень сітківки ока з використанням методів глибинного навчання, що дозволить здійснювати ефективну діагностику глаукоми та оцінювати ризик її розвитку. У рамках досягнення поставленої мети було проведено всебічний аналіз наявних методів комп’ютерної обробки медичних зображень, зосереджуючи увагу на сучасних підходах до діагностики глаукоми. На основі отриманих висновків обґрунтовано вибір архітектури згорткових нейронних мереж, зокрема U-Net, яка була адаптована для задач сегментації анатомічних структур сітківки. Розроблено алгоритми виявлення оптичного диска та виїмки, а також реалізовано методику обчислення ключових діагностичних параметрів, зокрема співвідношення cup-

to-disc [11] ratio (CDR), що має важливе значення для виявлення глаукоматозних змін.

У межах роботи також створено систему класифікації зображень за ступенем ризику розвитку глаукоми, яка ґрунтується на аналізі виявлених морфологічних змін. Для зручності практичного застосування, розроблено програмний інтерфейс користувача, що забезпечує інтерактивний аналіз зображень та наочну візуалізацію результатів. Ефективність запропонованої системи була підтверджена шляхом її тестування на відкритих наборах медичних даних, що дало змогу оцінити точність і надійність [34] роботи алгоритмів.

Об’єктом дослідження виступає процес автоматизованої обробки зображень сітківки для цілей діагностики офтальмологічних захворювань. У свою чергу, предметом є методи, алгоритми та програмні засоби, спрямовані на сегментацію структур сітківки та виявлення характерних ознак глаукоми на основі глибоких нейронних мереж.

Практичне значення результатів полягає у можливості інтеграції розробленої системи в сучасну офтальмологічну практику. Зокрема, її впровадження [46] в кабінетах лікарів, у рамках скринінгових програм та телемедичних платформ дозволить значно підвищити швидкість і точність аналізу зображень, забезпечити стандартизацію діагностичного процесу, а також розширити доступ до якісної медичної допомоги для мешканців віддалених територій.

З теоретичного боку дослідження робить внесок у розвиток напрямів комп’ютерного зору й машинного навчання, що застосовуються в медичній галузі. Особливу увагу приділено адаптації архітектури U-Net до специфіки офтальмологічних зображень, а також розробці комплексного підходу до оцінювання глаукомного ризику з використанням множинних біомаркерів.

Гіпотетично передбачається, що залучення глибоких нейронних мереж до аналізу зображень сітківки дає змогу досягти рівня точності діагностики,

порівняного з результатами досвідчених лікарів, водночас значно скорочуючи час обробки та підвищуючи ефективність діагностичного процесу.

Наукова новизна полягає у формуванні комплексної методики автоматизованої діагностики глаукоми, яка поєднує високоточну сегментацію анатомічних структур з подальшим аналізом клінічно важливих ознак. Крім того, у роботі запропоновано ефективний підхід до візуалізації [42] результатів, що забезпечує високу інтерпретованість рішень і сприяє їх інтеграції у клінічну практику.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд проблеми діагностики захворювань сітівки ока

Захворювання сітківки ока належать до найбільш поширених офтальмологічних патологій, що можуть призводити до незворотної втрати зору. Серед них особливе місце займають глаукома, діабетична ретинопатія та вікова макулодистрофія (AMD), які характеризуються тривалим безсимптомним перебігом на ранніх стадіях, що ускладнює їх своєчасну діагностику. За даними Всесвітньої організації охорони здоров’я, понад 76 мільйонів людей у світі страждають від глаукоми, і ця кількість продовжує зростати. Глаукома посідає провідне місце серед причин незворотної сліпоти у світі, при цьому близько 60% випадків захворювання залишаються невиявленими на ранніх стадіях.

Глаукома характеризується поступовим ураженням зорового нерва, що найчастіше зумовлено підвищенням внутрішньоочного тиску, з подальшою втратою периферійного, а згодом і центрального зору. Особливу небезпеку становить безсимптомний перебіг захворювання на ранніх стадіях, коли пацієнт не помічає поступового погіршення зору до моменту незворотних змін. Діабетична ретинопатія, яка виникає як ускладнення цукрового діабету, також має прогресуючий характер і може призводити до відшарування сітківки та сліпоти. Вікова макулодистрофія уражає центральну частину сітківки (макулу), що відповідає за найбільш чіткий зір, і є провідною причиною втрати зору серед людей старше 50 років у розвинених країнах.

Традиційні методи діагностики захворювань сітківки ґрунтуються на візуальному аналізі зображень очного дна (фундус-фотографії) та оптичної когерентної томографії (ОСТ) офтальмологами. Однак цей процес має кілька суттєвих обмежень, які знижують ефективність раннього виявлення патологій.

По-перше, ручний аналіз зображень є трудомістким і вимагає значних часових витрат кваліфікованого спеціаліста. Офтальмолог повинен детально

вивчити знімок, оцінити стан різних структур сітківки, зокрема оптичного диска, судинної мережі, макули, та виявити патологічні зміни, які можуть бути мінімальними на ранніх стадіях захворювання. В умовах постійно зростаючого навантаження на медичні заклади це призводить до обмеження кількості пацієнтів, яких може обстежити спеціаліст.

По-друге, існує проблема нерівномірного доступу до спеціалізованої офтальмологічної допомоги. У багатьох регіонах спостерігається дефіцит кваліфікованих офтальмологів, особливо в сільській місцевості та в країнах з низьким та середнім рівнем доходу. Це призводить до затримки діагностики та початку лікування, що особливо критично для захворювань, ефективність терапії яких значною мірою залежить від ранньої діагностики.

По-третє, суб’єктивність [5] оцінки зображень сітківки може призводити до варіабельності діагностичних висновків. Дослідження показують, що навіть досвідчені офтальмологи можуть по-різному інтерпретувати одні й ті ж знімки, особливо в неоднозначних або граничних випадках. Така міжекспертна варіабельність негативно впливає на точність діагностики та послідовність спостереження за прогресуванням захворювання.

З огляду на ці проблеми, розробка автоматизованих систем для аналізу зображень сітківки набуває особливої актуальності. Використання алгоритмів комп’ютерного зору та штучного інтелекту дозволяє не лише прискорити процес аналізу, але й підвищити його точність, забезпечити стандартизацію діагностичних критеріїв та розширити доступ до офтальмологічного скринінгу для населення віддалених районів. Такі системи особливо цінні для масового скринінгу, коли необхідно проаналізувати великі обсяги даних за короткий час, а також для телемедицини, що дозволяє проводити дистанційне консультування пацієнтів [31]. Сучасні технології штучного інтелекту, зокрема глибинні нейронні мережі, продемонстрували значний потенціал у автоматичній діагностиці захворювань сітківки. Вони здатні виявляти ранні ознаки патології, класифікувати стадії захворювання та прогнозувати ризик прогресування з точністю, порівнянною або навіть вищою за точність

діагностики досвідченими офтальмологами. Однак розробка таких систем стикається з певними викликами, зокрема необхідністю великих розмічених наборів даних для навчання, забезпеченням надійності та інтерпретованості результатів, а також інтеграцією в клінічну практику [47].

1.2 Аналіз методів комп’ютерної обробки зображень сітківки

Комп’ютерний аналіз зображень сітківки є ключовим компонентом автоматизованої діагностики офтальмологічних захворювань. Розвиток цього напрямку тісно пов’язаний з еволюцією методів комп’ютерного зору та машинного навчання, які пройшли шлях від традиційних алгоритмів обробки зображень до складних нейромережевих архітектур.

Основними типами зображень, що використовуються для діагностики захворювань сітківки, є фундус-фотографії, оптична когерентна томографія (ОСТ) та окулярна ангіографія (FA/OCT-A). Фундус-фотографії представляють собою кольорові знімки очного дна, які дозволяють оцінити стан оптичного диска, судин, макули та інших структур сітківки. Вони широко застосовуються у клінічній практиці завдяки відносній доступності обладнання та неінвазивності процедури. ОСТ забезпечує тривимірну візуалізацію шарів сітківки з мікронною роздільною здатністю, що дозволяє виявляти навіть мінімальні структурні зміни. Окулярна ангіографія, у свою чергу, дозволяє детально вивчити судинну мережу сітківки, що особливо важливо для діагностики діабетичної ретинопатії та інших судинних патологій.

Традиційні методи обробки зображень сітківки включають ряд послідовних етапів, спрямованих на покращення якості зображення, виділення анатомічних структур та виявлення патологічних ознак. На етапі попередньої обробки застосовуються фільтри для зменшення шуму (наприклад, фільтр Гауса або медіанний фільтр), методи корекції нерівномірного освітлення та

нормалізації контрасту. Ці процедури дозволяють підвищити якість вхідних даних для подальшого аналізу, особливо для зображень з низькою контрастністю або артефактами.

Наступним етапом є сегментація ключових анатомічних структур сітківки. Для виділення оптичного диска часто використовується трансформація Хафа, яка ефективно виявляє кругові об’єкти на зображенні. Сегментація судинної мережі може бути реалізована за допомогою методу активних контурів або морфологічних операцій. Детектор країв Кенні застосовується для виявлення меж між різними структурами сітківки. Такі алгоритми мають відносно низьку обчислювальну складність і не потребують великих навчальних наборів даних, однак їх ефективність знижується при аналізі зображень з патологічними змінами або низькою якістю.

Методи машинного навчання розширили можливості автоматизованого аналізу зображень сітківки, дозволяючи класифікувати патології та оцінювати їх тяжкість. Класичні алгоритми, такі як Support Vector Machines (SVM), Random Forest та k-Nearest Neighbors (k-NN), працюють з заздалегідь визначеними ознаками, виділеними з зображень. Ці ознаки можуть включати статистичні характеристики кольору та текстури, геометричні параметри анатомічних структур, показники форми та розміру об’єктів. Наприклад, для аналізу текстури часто використовуються Local Binary Patterns (LBP) та матриці співвзустрічей рівнів сірого кольору (Gray-Level Co-occurrence Matrix, GLCM).

Ефективність таких підходів сильно залежить від якості вибраних ознак і часто вимагає залучення експертних знань для їх визначення. Крім того, ці методи обмежені у можливостях виявлення складних взаємозв’язків між ознаками та мають тенденцію до зниження точності при аналізі гетерогенних даних.

Революційним проривом у комп’ютерному аналізі медичних зображень стало застосування глибинного навчання, зокрема згорткових нейронних мереж (Convolutional Neural Networks, CNN). На відміну від класичних

методів, CNN не потребують ручного виділення ознак, а автоматично навчаються виявляти ієрархічні патерни безпосередньо з пікселів зображення. Це особливо цінно для аналізу складних структур сітківки та різноманітних патологічних змін.

Сучасні архітектури CNN, такі як VGG16/VGG19, ResNet-50 та EfficientNet, продемонстрували високу ефективність у класифікації зображень сітківки. VGG-мережі мають просту послідовну структуру з декількома згортковими шарами, однак велика кількість параметрів (близько 138 мільйонів для VGG16) підвищує ризик перенавчання, особливо при обмежених навчальних даних. ResNet-50 вирішує проблему затухання градієнта у глибоких мережах за допомогою залишкових з'єднань (skip connections), що дозволяє будувати значно глибші архітектури. EfficientNet оптимізує співвідношення між глибиною, шириною та роздільною здатністю моделі, досягаючи вищої точності при меншій кількості параметрів.

Для задач сегментації структур сітківки особливу увагу заслуговує архітектура U-Net, яка має форму літери «U» з енкодером для зменшення просторової розмірності і декодером для її відновлення, з’єднаними skip-з’єднаннями. Ця архітектура ефективно поєднує контекстну інформацію з локалізацією об’єктів, що особливо важливо для точної сегментації оптичного диска, виїмки та судинної мережі.

Для виявлення локальних патологічних змін на зображеннях сітківки, таких як мікроаневризми, крововиливи або ексудати, успішно застосовуються алгоритми об’єктної детекції, зокрема YOLOv5 (You Only Look Once). Ці алгоритми здатні одночасно локалізувати та класифікувати патологічні зміни з високою точністю та швидкістю.

Особливий інтерес представляють універсальні моделі для виявлення очних та системних захворювань, такі як RETFound. Ці моделі навчаються на великих різноманітних наборах даних і здатні виявляти широкий спектр патологій, а також виявляти системні захворювання за змінами в сітківці,

наприклад, ознаки цукрового діабету, гіпертонії або серцево-судинних захворювань.

Сучасні підходи до аналізу зображень сітківки часто поєднують різні методи в єдиний конвеєр обробки даних. Наприклад, попередня обробка за допомогою традиційних алгоритмів може покращити якість вхідних даних для нейронних мереж, а сегментація ключових структур за допомогою U-Net може передувати класифікації патологій з використанням ResNet. Такі гібридні підходи дозволяють максимально використати переваги різних методів та підвищити загальну ефективність системи.

1.3 Огляд існуючих рішень та алгоритмів для виявлення глаукоми

Автоматизована діагностика глаукоми на основі аналізу зображень сітківки є активною областю досліджень, що призвело до розробки різноманітних алгоритмів та програмних рішень. Існуючі підходи можна класифікувати за типом аналізованих зображень, використовуваними алгоритмами та клінічними маркерами, на які вони орієнтуються.

Одним із перших значущих досліджень у цій області була робота Asaoka et al. (2019), в якій автори застосували глибинне навчання та трансферне навчання [17] для діагностики ранньої стадії глаукоми за знімками ОСТ макулярної області. Запропонований ними метод досяг чутливості 82,5% та специфічності 93,9%, що перевершило результати традиційних методів діагностики. Ключовою інновацією було використання даних ОСТ макули замість більш традиційного аналізу диска зорового нерва, що дозволило виявляти ранні зміни, які не завжди помітні при стандартному офтальмоскопічному обстеженні.

Система Medinoid, розроблена Kim [22] et al. (2019), представляє собою комплексне рішення для діагностики глаукоми та локалізації патологічних змін на фундус-зображеннях. Автори використали архітектуру глибокої

згорткової нейронної мережі на основі ResNet, яка не лише класифікує зображення як нормальні або глаукоматозні, але й формує теплову карту (heatmap), що вказує на ділянки, які найбільше вплинули на рішення моделі. Такий підхід забезпечує інтерпретованість результатів, що є важливим аспектом для клінічного застосування.

Значний внесок у розвиток автоматизованої діагностики глаукоми зробили An et al. [23] (2018), які порівняли різні моделі машинного навчання для класифікації зображень очного дна. Їхнє дослідження продемонструвало, що глибокі нейронні мережі перевершують традиційні методи машинного навчання, такі як SVM та Random Forest, особливо при обмеженому наборі навчальних даних. Автори також відзначили важливість попередньої обробки зображень для підвищення точності класифікації.

Liao et al. (2019) запропонували клінічно інтерпретовану модель глибокого навчання для діагностики глаукоми. Особливістю їхнього підходу є поєднання автоматичного виділення ознак нейронною мережею з клінічно значущими маркерами, такими як CDR (Cup-to-Disc Ratio) та RNFL (Retinal Nerve Fiber Layer). Це дозволило не лише досягти високої точності діагностики (AUC [19] 0,94), але й забезпечити зрозумілість результатів для офтальмологів, що сприяє довірі до системи.

Проект RIM-ONE [14] (An Open Retinal Image Database for Optic Nerve Evaluation) відіграв важливу роль у розвитку алгоритмів для виявлення глаукоми, надаючи дослідникам доступ до стандартизованого набору даних з анотаціями експертів. На основі цього датасету було розроблено кілька успішних моделей, включаючи роботу Gómez-Valverde et al. (2019), які застосували трансферне навчання для класифікації кольорових фундус-зображень. Їхня модель на основі DenseNet досягла точності 89,1% та специфічності 93,1%.

Важливим напрямком досліджень є точна сегментація оптичного диска та виїмки, що необхідно для розрахунку CDR. Yu [26] et al. (2019) запропонували робастний метод сегментації з використанням глибокого

навчання, який ефективно працює навіть з зображеннями низької якості. Їхній підхід базується на модифікованій архітектурі U-Net і включає етап попередньої локалізації області інтересу, що підвищує ефективність сегментації.

Kim et al. (2019) у своєму дослідженні також зосередились на сегментації оптичного диска та виїмки для характеристики глаукоми з використанням глибокого навчання. Вони порівняли різні архітектури нейронних мереж і показали, що модель на основі SegNet з використанням техніки атеншену (attention mechanism) забезпечує найкращі результати з коефіцієнтом Dice [18] 0,92 для диска та 0,87 для виїмки.

Серед комерційних рішень у галузі автоматизованої діагностики захворювань сітківки варто відзначити платформу RetinAI, яка використовує алгоритми штучного інтелекту для аналізу як фундус-зображень, так і ОСТ. Ця система здатна автоматично сегментувати структури сітківки, виявляти патологічні зміни та відстежувати прогресування захворювання в часі. RetinAI інтегрується з існуючими офтальмологічними інформаційними системами, що спрощує її впровадження в клінічну практику.

Інший приклад успішного впровадження штучного інтелекту в діагностику захворювань сітківки — система, розроблена спільно Inselspital та ARTORG Center. Вона фокусується на аналізі ОСТ-зображень та здатна передбачати попит на лікування хронічних захворювань очей, що допомагає оптимізувати планування ресурсів у клініках.

Дослідження Keel et al. (2019) представляє цікавий підхід до візуалізації процесу прийняття рішень глибокими моделями при діагностиці діабетичної ретинопатії та глаукоми. Автори використали градієнтно-зважені карти активації класів (Grad-CAM [20]) для ідентифікації регіонів зображення, які найбільше впливають на діагностичне рішення. Ця техніка дозволяє не лише покращити інтерпретованість моделей, але й виявити потенційні помилки або артефакти, що можуть вплинути на точність діагностики.

Спільною тенденцією серед сучасних досліджень є перехід від моделей загальної класифікації «норма/патологія» до більш детальної оцінки, включаючи виявлення стадії захворювання та ризику прогресування. Так, Akter et al. (2022) розробили систему мультимаркерн [38]ого аналізу та глибокого навчання для діагностики глаукоми, яка не лише класифікує зображення, але й оцінює стадію захворювання на основі множинних біомаркерів.

Важливим аспектом автоматизованої діагностики глаукоми є надійність роботи алгоритмів на різноманітних наборах даних. Sulot et al. (2021) запропонували метод ансамблевого глибокого навчання для класифікації глаукоми на основі зображень сканувальної лазерної офтальмоскопії. Їхній підхід об’єднує декілька моделей з різними архітектурами, що підвищує робастність системи при роботі з даними з різних джерел.

Для покращення ефективності моделей при обмежених навчальних даних Maheshwari & Kumar (2022) дослідили різні техніки аугментації даних на основі локальних дескрипторів для виявлення глаукоми. Вони показали, що застосування геометричних трансформацій у поєднанні з модифікаціями інтенсивності пікселів може значно підвищити точність моделей, особливо при малих навчальних наборах.

Серед найновіших розробок варто відзначити дослідження Cho et al. (2024), які використали механізм уваги (attention [39] mechanism) для класифікації глаукоми за зображеннями очного дна. Їхня модель фокусується на найбільш інформативних регіонах зображення, ігноруючи нерелевантні області, що підвищує точність діагностики та зменшує обчислювальні вимоги.

1.4 Формування вимог та постановка задачі дослідження

На основі проведеного аналізу предметної області, існуючих методів комп’ютерної обробки зображень сітківки та наявних рішень для діагностики

глаукоми, можна сформулювати ключові вимоги до розроблюваної системи автоматизованого аналізу сітківки для діагностики захворювань очей.

Функціональні вимоги до системи включають:

– завантаження та попередня обробка зображень сітківки різних форматів (фундус-фотографії, ОСТ) з можливістю корекції якості зображення;

– сегментація ключових анатомічних структур сітківки, зокрема оптичного диска, оптичної виїмки та судинної мережі;

– обчислення діагностичних параметрів, у першу чергу співвідношення cup-to-disc ratio (CDR), яке є критичним маркером для діагностики глаукоми;

– класифікація зображень з визначенням наявності патологічних змін та оцінкою ризику глаукоми;

– візуалізація результатів аналізу з виділенням відповідних структур на зображенні та відображенням кількісних параметрів.

Нефункціональні вимоги до системи:

– точність діагностики: система повинна забезпечувати високу чутливість (>85%) та специфічність (>90%) при виявленні глаукоми, що відповідає сучасним стандартам автоматизованої діагностики;

– швидкодія: час аналізу одного зображення не повинен перевищувати 5 секунд на стандартному комп’ютері, що дозволить ефективно використовувати систему в клінічних умовах;

– зручність використання: інтерфейс системи має бути інтуїтивно зрозумілим для медичного персоналу без спеціальної технічної підготовки;

– надійність: система повинна стабільно працювати з зображеннями різної якості та коректно обробляти граничні випадки;

– масштабованість: архітектура системи має передбачати можливість розширення функціональності, включаючи додавання нових методів аналізу та підтримку додаткових типів зображень;

– інтерпретованість результатів: система повинна не лише надавати діагностичний висновок, але й візуалізувати ключові фактори, що вплинули на це рішення, для підвищення довіри медичних спеціалістів;

– конфіденційність: система має забезпечувати захист персональних даних пацієнтів відповідно до чинного законодавства.

Для реалізації цих вимог планується використати сучасні методи глибинного навчання, зокрема архітектуру U-Net для сегментації структур сітківки та згорткові нейронні мережі (ResNet, EfficientNet) для класифікації патологічних змін. Для навчання моделей будуть використані відкриті датасети, такі як DRISHTI-GS [13], RIM-ONE та REFUGE, які містять розмічені зображення сітківки з анотаціями експертів.

На основі зазначених вимог можна сформулювати наступну постановку задачі дослідження: необхідно розробити програмний додаток для автоматизованого аналізу зображень сітківки з метою діагностики глаукоми та інших захворювань очей. Система повинна забезпечувати:

– автоматичну сегментацію оптичного диска та оптичної виїмки на зображеннях сітківки з високою точністю (коефіцієнт Dice >0,9);

– обчислення співвідношення cup-to-disc ratio (CDR) та інших клінічно значущих параметрів;

– класифікацію стану сітківки з визначенням ризику глаукоми на основі виявлених структурних змін;

– візуалізацію результатів аналізу з виділенням патологічних змін та представленням кількісних показників;

– зручний користувацький інтерфейс для завантаження зображень, перегляду та збереження результатів.

Результатом дослідження має бути функціональний програмний додаток, який зможе ефективно виявляти ранні ознаки глаукоми та інших захворювань сітківки, забезпечуючи високу точність діагностики та зручність використання в клінічних умовах. Система має потенціал для впровадження в офтальмологічних кабінетах, скринінгових програмах та телемедичних

системах, що сприятиме ранньому виявленню та своєчасному лікуванню офтальмологічних захворювань.

Таким чином, створення автоматизованої системи аналізу зображень сітківки є актуальною науково-практичною задачею, вирішення якої сприятиме підвищенню ефективності діагностики захворювань очей та покращенню якості офтальмологічної допомоги.

2 МАТЕМАТИЧНЕ ОБҐРУНТУВАННЯ ВИБРАНИХ МЕТОДІВ

2.1 Теоретичні основи згорткових нейронних мереж для аналізу медичних зображень

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) стали основним інструментом для аналізу медичних зображень, зокрема для діагностики офтальмологічних захворювань за зображеннями сітківки ока. Їх ефективність обумовлена здатністю автоматично вивчати ієрархічні ознаки безпосередньо з вхідних даних, без необхідності ручного проектування дескрипторів ознак. У цьому підрозділі розглянемо математичні основи CNN та їх архітектурних компонентів.

$$(F \times h)(\rho) = \sum_{s+t=p} F(s)h(t) = \sum_{s} F(s)h(p-s). \qquad (2.1)$$

Для двовимірних зображень ця операція розширюється до:

$$(I \times K)(i,j) = \sum m \sum n\, I(i - m, j - n)K(m,n), \qquad (2.2)$$

де I – вхідне зображення;

K – згортковий фільтр (ядро);

(i, j) – координати вихідного зображення.

У контексті глибокого навчання ця операція реалізується як:

$$Z^{(l)}i,i,k = \mathrm{Sm} \sum_n \sum_c W^{(l)}_{m,n,c,k} \cdot a^{(l-1)}_{i+m,j+n,c} + b^{(l)}_k, \qquad (2.3)$$

де $Z^{(l)}_{i,j,k}$ – результат згортки для позиції (i, j) та k-го фільтра в шарі l,

$W^{(l)}$ – ваги фільтрів;

$a^{(l-1)}$ – активації попереднього шару;

$b^{(l)}$ – вектор зміщення.

Після операції згортки застосовується нелінійна функція активації, найчастіше ReLU (Rectified Linear Unit):

$$ReLU(x) = max(0, x). \tag{2.4}$$

Ця функція вводить нелінійність у модель, дозволяючи навчатися більш складним патернам. ReLU має кілька переваг перед іншими функціями активації: вона обчислювально ефективна, не страждає від проблеми зникаючого градієнта при зворотному поширенні помилки та сприяє розрідженості активацій, що може покращити узагальнюючу здатність моделі.

Іншим ключовим компонентом CNN є операція підвибірки (pooling), яка зменшує просторові розміри представлення, зберігаючи найважливішу інформацію. Найпоширенішим варіантом є max-pooling, що математично описується як:

$$p^{i,j,k} = \max_{0 \le m < f, 0 \le n < f} a_{i \cdot s+m, j \cdot s+n, k}, \tag{2.5}$$

де f – розмір вікна підвибірки,

s – крок (stride);

a – вхідні активації;

p – вихід операції підвибірки.

Max-pooling зменшує чутливість вихідного представлення до незначних зсувів і дисторсій вхідних даних, що особливо корисно для аналізу медичних зображень, де можуть бути варіації в розташуванні анатомічних структур.

Процес навчання CNN включає мінімізацію функції втрат шляхом оптимізації ваг мережі за допомогою алгоритму зворотного поширення помилки (backpropagation) та методів оптимізації на основі градієнтного спуску. Для задачі класифікації медичних зображень часто використовується крос- ентропійна функція втрат:

$$L(y, \hat{y}) = -\sum i \; yilog(\hat{y}i), \tag{2.6}$$

де y – істинні мітки;

$\hat{y}$ – прогнозовані ймовірності.

Для задачі сегментації зображень часто використовують функцію втрат Dice, яку розглянемо детальніше в розділі 2.4.

Оновлення ваг мережі відбувається за правилом:

$$W(l)\, new = W(l)\, old - \eta \cdot \nabla \mu L, \tag{2.7}$$

де η – швидкість навчання;

$\nabla_W L$ – градієнт функції втрат відносно ваг.

За останнє десятиліття було розроблено кілька значущих архітектур CNN, які знайшли застосування в аналізі медичних зображень. Серед них варто відзначити AlexNet, VGG, ResNet і EfficientNet, які внесли важливі інновації в структуру згорткових мереж.

AlexNet, запропонована Krizhevsky та ін. у 2012 році, стала проривною архітектурою, яка включає п’ять згорткових шарів і три повнозв’язні шари. ЇЇ особливістю було використання ReLU замість традиційних на той час сигмоїдальних функцій активації, застосування dropout для запобігання перенавчанню та використання перекриваючихся пулінг-вікон.

Архітектура VGG, розроблена Simonyan та Zisserman з Оксфордського університету, характеризується використанням дуже малих (3×3) згорткових фільтрів з кроком 1 у всіх шарах мережі. Це дозволяє ефективно збільшувати глибину мережі, оскільки стек із двох 3×3 згорткових шарів має ефективне рецептивне поле 5×5, але з меншою кількістю параметрів. Важливо відзначити, що VGG16 і VGG19, попри свою відносну простоту, мають велику кількість параметрів (близько 138 мільйонів для VGG16), що може призводити до проблем з перенавчанням, особливо при аналізі медичних зображень, де кількість розмічених даних часто обмежена.

Архітектура ResNet, запропонована He та ін., вирішила проблему деградації точності у дуже глибоких мережах шляхом введення залишкових з'єднань (residual connections). Основною ідеєю ResNet є додавання вхідних даних до виходів згорткових шарів:

$$f(x) = F(x) + x, \tag{2.8}$$

де *H(x)* – бажане відображення;

F(x) – залишкова функція, яку вивчає набір шарів.

Це дозволяє градієнтам легше поширюватися через мережу, що значно полегшує навчання глибоких архітектур. ResNet-50, що містить 50 шарів, стала однією з найпопулярніших архітектур для задач комп'ютерного зору, включаючи аналіз медичних зображень.

EfficientNet, запропонована Tan і Le, використовує метод складеного масштабування для систематичного балансування глибини, ширини та роздільної здатності мережі. Математично це можна виразити як:

$$depth = \alpha \times \phi, width = \beta \times \phi, \tag{2.9}$$

$$resolution = \gamma^{\phi}, \quad (2.10)$$

де α, β, γ – коефіцієнти масштабування для кожного виміру, обмежені умовою, а φ – коефіцієнт складеного масштабування.

Цей підхід дозволяє EfficientNet досягати вищої точності з меншою кількістю параметрів порівняно з іншими архітектурами, що особливо важливо для медичних застосувань, де обчислювальні ресурси можуть бути обмеженими.

Застосування CNN для аналізу зображень сітківки вимагає врахування специфіки медичних зображень, зокрема:

– обмежена кількість розмічених даних, що вимагає ефективних стратегій регуляризації та аугментації;

– висока варіабельність між пацієнтами, що потребує робастних моделей;

– необхідність локалізації патологічних змін, що вимагає архітектур, здатних до точної сегментації;

– важливість інтерпретованості результатів для клінічного застосування.

Для вирішення цих проблем у медичному комп'ютерному зорі часто використовуються такі техніки, як трансферне навчання (transfer learning), аугментація даних, ансамблеві методи та методи пояснюваного штучного інтелекту (explainable AI).

Трансферне навчання передбачає використання моделі, попередньо навченої на великому наборі даних (наприклад, ImageNet), з подальшим донавчанням на специфічних медичних зображеннях. Математично це означає ініціалізацію ваг мережі вагами попередньо навченої мережі і їх подальше налаштування:

$$W_{target} = Fine - tune\ (W_{source}, D_{target}), \quad (2.11)$$

де D_{target} – цільовий набір медичних даних.

Цей підхід дозволяє ефективно використовувати знання, отримані з великих наборів даних загального призначення, для вирішення спеціалізованих медичних задач.

2.2 Архітектура U-Net для сегментації зображень сітківки

Архітектура U-Net, запропонована Ronneberger, Fischer та Brox у 2015 році, стала одним із найефективніших інструментів для сегментації медичних зображень, включаючи зображення сітківки ока. Її ключовою особливістю є симетрична структура у формі літери «U», яка складається з енкодера (шлях стиснення) та декодера (шлях розширення), з'єднаних за допомогою skip-з'єднань.

Математично процес кодування в U-Net можна описати як послідовність операцій згортки та підвибірки:

$$E_k = Pool\,(ReLU(Conv(E_{k-1}))), \tag{2.12}$$

де E_k – вихід k-го рівня енкодера;

Conv – операція згортки;

ReLU – функція активації;

Pool – операція підвибірки (зазвичай max-pooling).

Процес декодування включає операції розгортання (up-convolution або transpose convolution) та конкатенацію з відповідними картами ознак з енкодера:

$$D_k = ReLU\ (Conv(Concat(UpConv(D_{k+1}), E_k))), \quad (2.13)$$

де D_k – вихід k-го рівня декодера;

UpConv – операція розгортання;

Concat – операція конкатенації;

Важливим елементом архітектури U-Net є skip-з'єднання, які передають карти ознак з енкодера безпосередньо до відповідних шарів декодера через операцію конкатенації. Це дозволяє зберегти просторову інформацію високої роздільної здатності, яка втрачається при операціях підвибірки, і передати її у декодер для більш точної локалізації об'єктів. Математично це можна описати як:

$$F_{decoder,i} = Concat(F_{upsampled,i}, F_{encoder,i}), \quad (2.14)$$

де $F_{decoder,i}$ – вхідні карти ознак для і-го шару декодера;

$F_{upsampled,i}$ – розгорнуті карти ознак з попереднього шару декодера;

$F_{encoder,i}$ – карти ознак з відповідного шару енкодера.

Архітектура U-Net, реалізована для сегментації структур сітківки, складається з наступних компонентів:

Енкодер включає послідовні блоки подвійної згортки, кожен з яких містить дві згортки з ядром 3×3, за якими йде операція batch-нормалізації та функція активації ReLU:

$$DoubleConv(x) = ReLU\ (BN\ (Conv3 \times 3\ (ReLU\ (BN(Conv3 \times 3(x)))))), \quad (2.15)$$

де *BN* – операція batch-нормалізації.

Після кожного блоку подвійної згортки (крім останнього) виконується операція max-pooling з вікном 2×2, що зменшує просторові розміри вдвічі:

$$Down(x) = DoubleConv(MaxPool_{2\times2}(x)). \quad (2.16)$$

Це призводить до послідовного зменшення просторових розмірів і збільшення глибини каналів:

– вхід: 512×512×3 (для кольорового зображення)
– рівень 1 енкодера: 256×256×64;
– рівень 2 енкодера: 128×128×128;
– рівень 3 енкодера: 64×64×25;
– рівень 4 енкодера: 32×32×512;
– найнижчий рівень (bottleneck): 16×16×1024.

Декодер, в свою чергу, включає послідовні блоки розгортання, які збільшують просторові розміри і зменшують глибину каналів. Кожен блок включає операцію розгортання (up-convolution), конкатенацію з відповідними картами ознак з енкодера і подвійну згортку:

$$Up(x, skip) = DoubleConv(Concat(UpConv2\times2(x), skip)), \quad (2.17)$$

де *UpConv* – операція розгортання, яка збільшує просторові розміри вдвічі;

skip – карти ознак з відповідного рівня енкодера.

Це призводить до послідовного збільшення просторових розмірів і зменшення глибини каналів:

– рівень 4 декодера: 32×32×512;
– рівень 3 декодера: 64×64×256;
– рівень 2 енкодера: 128×128×128;
– рівень 1 декодера: 256×256×64;

– Вихід: 512×512×N, де N — кількість класів для сегментації (у нашому випадку N=3: фон, оптичний диск, оптична виїмка).

На фінальному етапі застосовується згортка 1×1, яка перетворює вихідні карти ознак у канали сегментації:

$$Out(x) = Conv1 \times 1(x). \quad (2.18)$$

За цим слідує функція активації Sigmoid, яка нормалізує вихідні значення до діапазону [0, 1], що можна інтерпретувати як ймовірність належності кожного пікселя до відповідного класу:

$$\sigma(z)j = \frac{e^{z_j}}{ke^{zk}}. \quad (2.19)$$

У реалізації U-Net для сегментації структур сітківки, представленій у вихідному коді, використовуються певні модифікації, які покращують ефективність моделі:

– використання batch-нормалізації після кожної згортки, що прискорює збіжність навчання і зменшує залежність від ініціалізації ваг;

– застосування dropout з ймовірністю 0.5 після bottleneck шару для запобігання перенавчанню;

– використання біопеції для нормалізації вихідних даних.

Архітектура U-Net особливо ефективна для сегментації зображень сітківки з кількох причин:

– здатність працювати з обмеженою кількістю навчальних зразків завдяки інтенсивній аугментації даних;

– висока точність локалізації завдяки skip-з’єднанням, які зберігають просторову інформацію;

– можливість сегментації структур різного розміру завдяки мультимасштабному аналізу, що особливо важливо для виділення оптичного диску та виїмки, які можуть варіюватися за розміром;

– здатність розрізняти тонкі границі між класами, що критично для точного визначення меж оптичного диску та виїмки.

Для задачі сегментації структур сітківки, модель U-Net навчається мінімізувати функцію втрат Dice, яка забезпечує баланс між точністю та повнотою сегментації:

$$L_{\text{Dice}} = 1 - \frac{L.|X \cap Y| + s}{|X| + |Y| + s}, \tag{2.20}$$

де X – прогнозована маска;

Y – істинна маска;

ε – мала константа для чисельної стабільності.

Ця функція втрат особливо ефективна для задач з незбалансованими класами, що характерно для медичних зображень, де патологічні області часто займають малу частину зображення.

2.3 Методи виявлення діагностичних міток глаукоми

Виявлення діагностичних міток глаукоми на зображеннях сітківки є критичним етапом у автоматизованій діагностиці цього захворювання. Ключовими біомаркерами глаукоми, які можна ідентифікувати на фундус-зображеннях, є специфічні структурні зміни в області оптичного диска, витончення шару нервових волокон сітківки (RNFL [41]) та аномалії в конфігурації судинної мережі.

Найважливішим маркером для ранньої діагностики глаукоми є співвідношення діаметра оптичної виїмки до діаметра оптичного диска (Cup-

to- Disc Ratio, CDR). У здорової людини значення CDR зазвичай не перевищує 0.3- 0.4, тоді як значення понад 0.6 вважається підозрілим на глаукому. Математично CDR розраховується як:

$$CDR = \frac{\text{Площа оптичної виїмки}}{\text{Площа оптичного диска}}. \quad (2.21)$$

або, альтернативно, як:

$$CDR = \frac{\text{Діаметр оптичної виїмки}}{\text{Діаметр оптичного диска}}. \quad (2.22)$$

Для точного обчислення CDR необхідна сегментація оптичного диска та оптичної виїмки з максимальною точністю. Використовуючи сегментовані маски, отримані за допомогою архітектури U-Net, описаної в попередньому підрозділі, можна розрахувати площі цих структур за формулами:

$$A_{\text{disc}} = \sum M_{\text{disc}}(i,j), \quad (2.23)$$

$$A_{\text{cupp}} = \sum_{i,j} M_{\text{cup.}}(i,j), \quad (2.24)$$

де M_{disc} і M_{cup} – бінарні маски оптичного диска та оптичної виїмки відповідно;

(i,j) – координати пікселів.

Для більш стабільного обчислення CDR також можуть використовуватися морфологічні характеристики сегментованих структур. Наприклад, можна знайти найбільший і найменший діаметри оптичного диска та виїмки, або використовувати вертикальний CDR, який розраховується як:

$$\text{VCDR} = \frac{\text{Вертикальний діаметр виїмки}}{\text{Вертикальний діаметр диска}}. \quad (2.25)$$

Цей показник вважається більш інформативним, оскільки глаукоматозні зміни часто проявляються як вертикальне розширення оптичної виїмки.

Важливим фактором у ранній діагностиці глаукоми є асиметрія CDR між очима. Різниця CDR між правим і лівим оком, що перевищує 0.2, вважається діагностично значущою і вимагає додаткового обстеження. Розрахунок асиметрії CDR здійснюється за формулою:

$$\Delta CDR = |CDR_{right} - CDR_{left}|. \quad (2.26)$$

Крім CDR, важливим маркером глаукоми є стан шару нервових волокон сітківки (RNFL). У пацієнтів із глаукомою спостерігається прогресивне витончення RNFL, особливо в верхньому та нижньому квадрантах диска зорового нерва. Для кількісної оцінки товщини RNFL використовуються дані ОСТ, які дозволяють вимірювати товщину цього шару з мікронною точністю.

З фундус-зображень також можна отримати інформацію про стан RNFL шляхом аналізу текстури та яскравості в перипапілярній області (область навколо оптичного диска). Зміни в текстурі можуть бути кількісно оцінені за допомогою текстурних дескрипторів, таких як Local Binary Patterns (LBP) або матриця співзустрічей рівнів сірого (GLCM).

LBP оператор порівнює значення центрального пікселя з його сусідами і формує бінарний код, який потім перетворюється на десяткове число:

$$LBPP,R(x_c, y_c) = \sum p = 0P - 1s(g_p - g_c)2p, \quad (2.27)$$

де g_c – значення центрального пікселя;

g_p – значення p-го сусіда на відстані *R;*

g_p *s(x)* – функція знаку:

$$s(x) = [[1, \&x \geq 00, \&x < 0]]. \quad (2.28)$$

Матриця GLCM, в свою чергу, підраховує частоту зустрічі пар пікселів з певними значеннями інтенсивності, розташованих на заданій відстані та напрямку. На основі GLCM можна обчислити такі текстурні характеристики, як контраст, енергія, однорідність та кореляція:

$$\text{Контраст} = \sum_{j,i} |i - j|2 \cdot p(i,j), \quad (2.29)$$

$$\text{Енергія} = \sum_{j,i} p(i,j)^2, \quad (2.30)$$

$$\text{Однорідність} = \sum_{i,j} \frac{p(i,j)}{1+|i-j|}, \quad (2.31)$$

$$\text{Кореляція} = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_i)p(i,j)}{\sigma_i \sigma_i}, \quad (2.32)$$

де *p(i,j)* – елемент матриці GLCM;

μ_i , μ_j , σ_i , σ_j – середні значення та стандартні відхилення.

Ще одним важливим діагностичним маркером глаукоми є аномалії в структурі судинної мережі сітківки. У пацієнтів з глаукомою спостерігаються такі зміни, як вигинання судин (bayonetting), назалізація судин (nasalization) та зміна кута відгалуження судин від центральної судинної аркади.

Для кількісної оцінки цих змін використовуються методи скелетизації судинної мережі з подальшим аналізом її топології. Скелетизація судинної мережі може бути виконана за допомогою морфологічних операцій або спеціалізованих алгоритмів трасування судин. У результаті формується граф,

вершинами якого є точки розгалуження та кінцеві точки судин, а ребрами — самі судини.

Аналіз цього графа дозволяє обчислити такі кількісні характеристики, як кути розгалуження судин, щільність судинної мережі, коефіцієнт звивистості судин та величину архаїчної медіанної артеріальної аркади. Зокрема, коефіцієнт звивистості судин розраховується як:

$$\text{Tortuosity} = \frac{L_{\text{actual}}}{L_{\text{straight}}}. \tag{2.33}$$

де L_{actual} – фактична довжина судини вздовж її осьової лінії;

L_{straight} – довжина прямої лінії, що з'єднує кінцеві точки судини.

Окрім структурних змін, для діагностики глаукоми важливо враховувати і кольорові характеристики оптичного диска. При глаукомі спостерігається поступова атрофія нервових волокон, що призводить до блідості диска зорового нерва. Ця зміна може бути кількісно оцінена шляхом аналізу кольорових гістограм в області оптичного диска та порівняння з нормативними значеннями. Для комплексної оцінки ризику глаукоми доцільно використовувати інтегральний показник, який враховує всі виявлені маркери. Одним із підходів є розрахунок ймовірності глаукоми на основі логістичної регресії:

$$P(\text{ Глаукома }) = \frac{1}{1 + e^{-(\beta + \beta \cdot \text{CDR} + \beta^2 \cdot \Delta\text{CDR} + \beta^3 \cdot \text{RNFL} + \beta^4 \text{ Vascular } + \beta^5 \text{ Color })}}, \tag{2.34}$$

де β_i – коефіцієнти моделі, які визначаються шляхом навчання на розміченому наборі даних;

RNFL – показник стану шару нервових волокон;

Vascular – показник стану судинної мережі;

Color – показник кольорових характеристик.

На основі розрахованого значення CDR можна провести класифікацію зображення за ризиком глаукоми. В реалізованій системі використовується наступна шкала:

– CDR < 0.4: норма (мінімальний ризик глаукоми);

– 0.4 ≤ CDR < 0.6: підозра на глаукому (низький ризик);

– 0.6 ≤ CDR < 0.75: помірний ризик глаукоми;

– 0.75 ≤ CDR < 0.85: високий ризик глаукоми;

– CDR ≥ 0.85: дуже високий ризик глаукоми, потребує негайного обстеження.

Важливим аспектом алгоритму виявлення діагностичних міток є візуалізація результатів для клінічної інтерпретації. Для цього використовується накладання сегментованих масок на оригінальне зображення з різними кольорами для оптичного диска (зазвичай зелений) та оптичної виїмки (зазвичай червоний). Крім того, для пояснення рішення моделі можуть використовуватися теплові карти (heatmaps), створені за допомогою техніки Gradient-weighted Class Activation Mapping (Grad-CAM).

Метод Grad-CAM дозволяє візуалізувати регіони зображення, які найбільше вплинули на рішення моделі, шляхом обчислення градієнтів вихідного класу відносно карт ознак останнього згорткового шару:

$$\alpha_k^c = \frac{1}{Z} \cdot \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \tag{2.35}$$

де y^c – прогнозований скор для класу c,

A^k – k-та карта ознак;

Z — нормалізуючий фактор. Теплова карта обчислюється як зважена сума карт ознак:

$$L^{c}_{\text{Grad-CAM}} = \text{ReLU}(2_k \alpha_k^c \cdot A^{\kappa}). \quad (2.36)$$

Ця візуалізація допомагає офтальмологам інтерпретувати результати автоматизованої діагностики та приймати більш обґрунтовані клінічні рішення

.

2.4 Оцінка ефективності та метрики сегментації медичних зображень

Оцінка ефективності алгоритмів сегментації медичних зображень є критичним етапом розробки автоматизованих систем діагностики. Для забезпечення клінічної цінності таких систем необхідно використовувати адекватні метрики, які враховують специфіку медичних зображень та клінічні вимоги до точності діагностики.

Базовими метриками для оцінки бінарної сегментації є точність (accuracy), чутливість (sensitivity), специфічність (specificity) та F1-міра. Вони обчислюються на основі матриці помилок (confusion matrix), яка включає такі компоненти:

– True Positive (TP): кількість пікселів, правильно класифікованих як об’єкт інтересу;

– True Negative (TN): кількість пікселів, правильно класифікованих як фон;

– False Positive (FP): кількість пікселів фону, неправильно класифікованих як об’єкт інтересу;

– False Negative (FN): кількість пікселів об’єкту інтересу, неправильно класифікованих як фон.

Точність (accuracy) представляє собою частку правильно класифікованих пікселів від загальної кількості пікселів:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}. \quad (2.37)$$

Чутливість (sensitivity) або повнота (recall) відображає здатність алгоритму виявляти всі пікселі, що належать до об’єкту інтересу:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}}. \quad (2.38)$$

Специфічність (specificity) показує здатність алгоритму правильно відхиляти пікселі, що не належать до об’єкту інтересу:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}. \quad (2.39)$$

Precision (точність у вузькому сенсі) вимірює, наскільки точно алгоритм ідентифікує пікселі об’єкту інтересу без помилкових спрацьовувань:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \quad (2.40)$$

F1-міра є гармонійним середнім між точністю (precision) та повнотою (recall), що забезпечує збалансовану оцінку:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (2.41)$$

Однак для задач сегментації медичних зображень ці метрики мають певні обмеження, особливо при значному дисбалансі класів, який характерний для багатьох патологій, включаючи структури сітківки, де об’єкт інтересу (наприклад, оптична виїмка) займає малу частину загального зображення.

Тому для оцінки сегментації медичних зображень часто використовують метрики на основі перекриття (overlap-based) та відстані (distance-based). Найпопулярнішою метрикою на основі перекриття є коефіцієнт Dice (Dice Similarity Coefficient, DSC) або F1-міра для сегментації:

$$\mathrm{DSC} = \frac{2\cdot|X\cap Y|}{|X|+|Y|} = \frac{2\cdot\mathrm{TP}}{2\cdot\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}, \tag{2.42}$$

де X та Y – прогнозована та істинна маски відповідно.

Коефіцієнт Dice має діапазон значень від 0 до 1, де 1 відповідає ідеальній сегментації.

Іншою популярною метрикою є коефіцієнт Жаккара (Jaccard Index) або Intersection over Union (IoU):

$$\mathrm{IoU} = \frac{|X\cap Y|}{|X\cup Y|} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}, \tag{2.43}$$

Коефіцієнт IoU також має діапазон від 0 до 1, де вищі значення відповідають кращій сегментації. Між коефіцієнтами Dice та Jaccard існує пряме співвідношення:

$$DSC = \frac{2\cdot IoU}{1+IoU}. \tag{2.44}$$

$$\mathrm{IoU} = \frac{\mathrm{DSC}}{2-\mathrm{DSC}}. \tag{2.45}$$

Для оцінки відстані між межами сегментованого об’єкта та еталонної маски часто використовується середня помилка поверхні (Average Surface Distance, ASD) та відстань Хаусдорфа (Hausdorff Distance, HD).

Середня помилка поверхні обчислюється як середня відстань від кожної точки межі сегментованого об’єкта до найближчої точки межі еталонної маски:

$$\mathrm{ASD}(X,Y) = \frac{1}{|B_X|} \cdot \sum_{x \in B_X} \min_{y \in B_Y} d(x,y), \tag{2.46}$$

де B_Xта B_Y – межі сегментованого та еталонного об’єктів відповідно;

d(x, y) — евклідова відстань між точками x та y.

Відстань Хаусдорфа представляє собою максимальну відстань від точки в одній множині до найближчої точки в іншій множині:

$$\mathrm{HD}(X,Y) = max\left(\max_{x \in B_X} \min_{y \in B_Y} d(x,y), \max_{y \in B_Y} \min_{x \in B_X} d(x,y)\right). \tag{2.47}$$

Відстань Хаусдорфа є більш чутливою до викидів, тому часто використовується її 95-й перцентиль (95HD), який ігнорує 5% найбільш віддалених точок.

Для специфічної задачі сегментації оптичного диска та оптичної виїмки з метою діагностики глаукоми важливою метрикою є точність обчислення CDR. Ця метрика оцінює різницю між значеннями CDR, обчисленими на основі сегментованих масок та еталонних анотацій:

$$CDRerror = | \ CDRpred - CDR\,true \ | . \tag{2.48}$$

Для клінічної валідації алгоритму важливо також оцінити кореляцію між автоматично обчисленими значеннями CDR та значеннями, визначеними експертами-офтальмологами. Для цього використовуються коефіцієнт кореляції Пірсона:

$$r = \frac{\text{cov}(\text{ CDRpred CDRtrue })}{\sigma_{\text{pred}} \cdot \sigma_{\text{true}}}. \quad (2.49)$$

Та аналіз Бленда-Альтмана, який дозволяє оцінити систематичну похибку та межі узгодження між двома методами вимірювання.

Для оцінки класифікаційної здатності моделі (тобто здатності відрізняти глаукоматозні зображення від нормальних) використовується аналіз ROC-кривої (Receiver Operating Characteristic) та обчислення площі під кривою (Area Under the Curve, AUC). ROC-крива будується шляхом відкладання чутливості (sensitivity) по осі Y та 1-специфічності (1-specificity) по осі X для різних порогових значень прогнозованого ризику:

$$\text{ROC} = (1 - \text{Specificity}(t), \text{Sensitivity}(t)) \mid t \in T\#. \quad (2.50)$$

де T — множина порогових значень.

AUC обчислюється як площа під ROC-кривою і варіюється від 0.5 (випадкова класифікація) до 1 (ідеальна класифікація).

В реалізованій системі для оцінки ефективності сегментації використовуються коефіцієнт Dice та середня абсолютна похибка CDR. Зокрема, для оцінки сегментації оптичного диска та оптичної виїмки окремо обчислюються:

$$\text{Dicedisc} = \frac{2 \cdot |X \text{ disc} \cap Y_{\text{disc}}|}{|X_{\text{disc}}| + |Y_{\text{disc}}|}, \quad (2.51)$$

$$\text{Dicecup} = \frac{2 \cdot |X \text{ cup} \cap Y_{\text{cup}}|}{|X_{\text{cup}}| + |Y_{\text{cup}}|}, \quad (2.52)$$

де X_{disc} та Y_{disc} – прогнозована та істинна маски оптичного диска;

X_{cup} та Y_{cup} – прогнозована та істинна маски оптичної виїмки.

Під час навчання моделі U-Net використовується функція втрат на основі коефіцієнта Dice, яка математично описується як:

$$L_{\text{Dice}} = 1 - \frac{1}{C} \cdot \sum_{c=1}^{C} \text{DSC}_c \,, \tag{2.53}$$

де C – кількість класів (у нашому випадку C=3: фон, оптичний диск, оптична виїмка), – коефіцієнт *Dice* для с-го класу.

Для стабілізації навчання часто використовується модифікована функція втрат з малим додатковим параметром ε:

$$L_{\text{Dice}} = 1 - \frac{1}{C} \cdot \sum_{c=1}^{C} \frac{2 \cdot |X_c \cap Y_c| + \varepsilon}{|X_c| + |Y_c| + \varepsilon}. \tag{2.54}$$

У реалізації проекту також використовується техніка раннього зупинення (early stopping) навчання, яка припиняє процес, коли валідаційна метрика (у нашому випадку — коефіцієнт Dice) перестає покращуватися протягом заданої кількості епох. Це допомагає запобігти перенавчанню моделі. Математично це можна описати як:

$$Early\ stopping \text{ на епосі } t, \text{якщо:} \tag{2.55}$$

$$\forall i \in 1,2, \dots, p: Diceval\,(t) \leq Diceval\,(t - i), \tag{2.56}$$

де p – параметр терпіння (patience), який визначає кількість епох без покращення метрики, після яких навчання буде зупинено.

Для валідації ефективності системи в цілому проводиться оцінка точності класифікації зображень за ризиком глаукоми, порівнюючи результати автоматичної діагностики з експертними оцінками. Ця оцінка

включає розрахунок точності, чутливості, специфічності та AUC для бінарної класифікації «норма/глаукома», а також матриці помилок для багатокласової класифікації за рівнями ризику.

Комплексний підхід до оцінки ефективності сегментації та класифікації забезпечує повне розуміння сильних сторін та обмежень розробленої системи, що важливо для її подальшого вдосконалення та клінічного впровадження.

3 РЕАЛІЗАЦІЯ ПОСТАНОВКИ ЗАДАЧІ ТА ТЕСТУВАННЯ

3.1 Архітектура програмного додатку

Для реалізації системи автоматизованого аналізу зображень сітківки розроблено програмний додаток з модульною архітектурою, що забезпечує розділення функціональності та зручність підтримки. Структура проекту представлена на рис. 3.1., де показано основні модулі та їх взаємозв'язки.

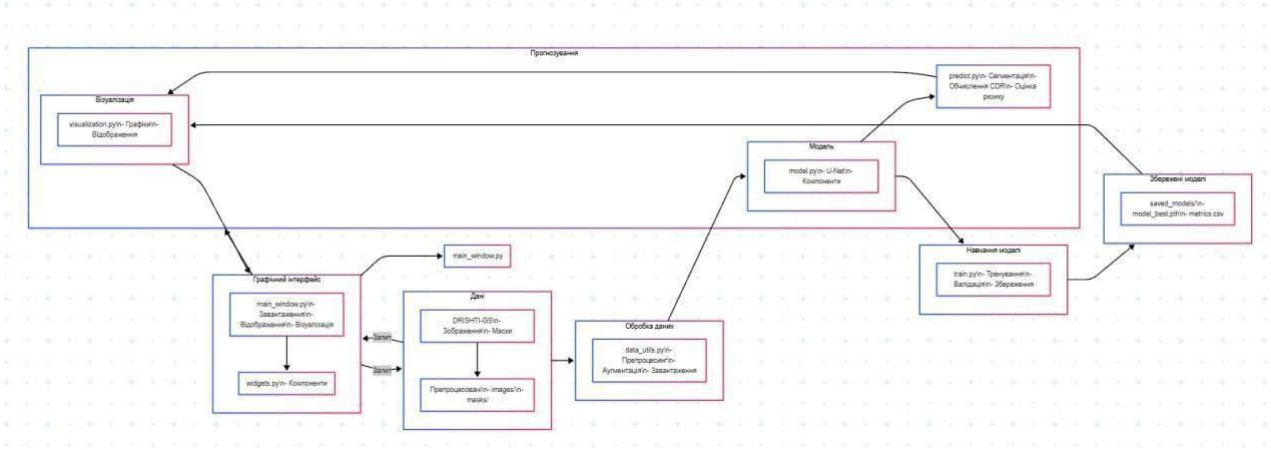

Рисунок 3.1 – Структура проєкту

Розроблена система організована навколо наступної ієрархії каталогів та модулів:

Каталог data/ призначений для зберігання наборів даних, включаючи оригінальні зображення з набору DRISHTI-GS та препроцесовані дані для тренування і тестування мережі.

Каталог src/ містить основні функціональні модулі:

– config.py — визначає глобальні константи та параметри системи;
– data_utils.py — реалізує функції обробки та підготовки даних;
– model.py — містить реалізацію архітектури U-Net;
– predict.py — функції для прогнозування та аналізу зображень;
– train.py — реалізує процес навчання нейронної мережі.

Каталог utils/ включає допоміжні модулі:

– metrics.py — реалізація метрик для оцінки якості моделі;

– visualization.py — набір функцій для візуалізації результатів.

Каталог gui/ містить модулі для реалізації графічного інтерфейсу:

– main_window.py — головне вікно програми;

– widgets.py — спеціалізовані віджети інтерфейсу.

Каталог saved_models/ зберігає навчені моделі з відповідними метриками.

Основою архітектури є принцип інверсії залежностей, де високорівневі модулі не залежать від низькорівневих, а обидва залежать від абстракцій. Це реалізовано через використання імпортів та чітке розділення інтерфейсів між компонентами.

Система реалізована з використанням Python 3.8+ та наступних основних бібліотек:

– PyTorch — для реалізації та навчання нейронних мереж;

– NumPy та Pandas — для обробки та аналізу даних;

– OpenCV — для операцій з зображеннями;

– PyQt [28]5 — для створення графічного інтерфейсу;

– Matplotlib — для візуалізації результатів.

Лістинг 3.1 Основні конфігураційні параметри системи визначені в модулі config.py:

 # Шляхи до даних

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(file)))
DATA_DIR = os.path.join(BASE_DIR, ‘data')

DRISHTI_DIR = os.path.join(DATA_DIR, 'DRISHTI-GS')

PREPROCESSED_DIR = os.path.join(DATA_DIR, 'preprocessed')

MODELS_DIR = os.path.join(BASE_DIR, 'saved_models')

 # Параметри моделі та тренування

IMAGE_SIZE = (512, 512) # Розмір для ресайзу зображень BATCH_SIZE = 4

EPOCHS = 30

LEARNING_RATE = 1e-4 VALIDATION_SPLIT = 0.2

 # Поріг для бінарної класифікації THRESHOLD = 0.5

 # Класи сегментації CLASSES = {

'background': 0,

'optic_disc': 1,

'cup': 2

}

 # Параметри для PyTorch

DEVICE = "cuda" if torch.cuda.is_available() else "cpu"

NUM_WORKERS = 4 # Кількість робочих потоків для завантаження даних

Такий підхід до конфігурації дозволяє легко налаштовувати систему без необхідності змінювати основний код, що відповідає принципам чистої архітектури

3.2 Підготовка та обробка даних

Якість даних та їх правильна підготовка є ключовими факторами для успішного навчання нейронної мережі. У розробленій системі реалізовано комплексний конвеєр обробки даних, який включає завантаження, препроцесинг, аугментацію та формування наборів для тренування та тестування.

Для навчання та тестування моделі використовується набір даних DRISHTI-GS, який містить 101 фундус-зображення сітківки з розміткою оптичного диска та оптичної виїмки. Для забезпечення якісного навчання

моделі необхідно правильно організувати ці дані та підготувати їх до використання в нейронній мережі.

Процес підготовки даних реалізовано в модулі data_utils.py і складається з наступних етапів:

Лістинг 3.2 Попередня обробка зображень та масок. Ця функція обробляє оригінальні зображення та відповідні маски, приводячи їх до стандартного розміру та формату:

```
def preprocess_dataset():
""" Функція для підготовки та попередньої обробки набору даних DRISHTI-GS відповідно до структури папок користувача """
print("Початок попередньої обробки датасету DRISHTI-GS...")
# Створення директорій для оброблених даних os.makedirs(os.path.join(PREPROCESSED_DIR,'images', 'train'),
exist_ok=True)
os.makedirs(os.path.join(PREPROCESSED_DIR, 'images', 'test'), exist_ok=True)
os.makedirs(os.path.join(PREPROCESSED_DIR, 'masks', 'train'), exist_ok=True)
os.makedirs(os.path.join(PREPROCESSED_DIR, 'masks', 'test'), exist_ok=True)
# Отримання списку тестових папок (вони містять ground truth) test_folders = [f for f in os.listdir(TEST_GT_DIR) if
os.path.isdir(os.path.join(TEST_GT_DIR, f))]
test_image_ids = [folder for folder in test_folders]
# Обробка тестових зображень з ground truth for image_id in test_image_ids:
process_test_gt_image(image_id)
# Визначаємо, які зображення йдуть у тренувальний набір normal_images = [f for f in os.listdir(NORMAL_DIR) if f.endswith('.jpg') or
```

```
f.endswith('.png')]
glaucoma_images = [f for f in os.listdir(GLAUCOMA_DIR) if f.endswith('.jpg') or f.endswith('.png')]
# Відфільтровуємо тестові зображення з тренувального набору train_normal_images = []
train_glaucoma_images = []
for img in normal_images:
image_id = img.split('_')[0] + '_' + img.split('_')[1] # Отримуємо drishtiGS_XXX
if image_id not in test_image_ids: train_normal_images.append(img)
for img in glaucoma_images:
image_id = img.split('_')[0] + '_' + img.split('_')[1] # Отримуємо drishtiGS_XXX
if image_id not in test_image_ids: train_glaucoma_images.append(img)
# Обробка зображень для тренування
for img in train_normal_images: process_train_image(img, 'normal', False)
for img in train_glaucoma_images: process_train_image(img, 'glaucoma', True)
print("Попередня обробка датасету завершена!")
```

Лістинг 3.3 Обробка тренувальних зображень. Функція process_train_image виконує обробку кожного тренувального зображення: завантаження, зміна розміру та створення відповідних масок:

```
def process_train_image(filename, image_type, is_glaucoma): try:
# Шлях до вхідного зображення if image_type == 'normal':
img_path = os.path.join(NORMAL_DIR, filename) else:
img_path = os.path.join(GLAUCOMA_DIR, filename)
# Завантаження та обробка зображення img = cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
img = cv2.resize(img, IMAGE_SIZE, interpolation=cv2.INTER_AREA)
# Отримання ID зображення
base_filename = os.path.splitext(filename)[0] parts = base_filename.split('_')
if len(parts) >= 2:
image_id = parts[0] + '_' + parts[1] else:
image_id = base_filename
# Створення маски
mask = np.zeros((*IMAGE_SIZE, 3), dtype=np.uint8)
# Збереження зображення та маски
out_img_path = os.path.join(PREPROCESSED_DIR, 'images', 'train', f"{image_id}.png")
out_mask_path = os.path.join(PREPROCESSED_DIR, 'masks', 'train', f"{image_id}.png")
cv2.imwrite(out_img_path, cv2.cvtColor(img, cv2.COLOR_RGB2BGR)) cv2.imwrite(out_mask_path, mask)
except Exception as e:
print(f"Помилка при обробці зображення {filename}: {e}")
```

Лістинг 3.4 Аугментація даних. Для збільшення різноманітності тренувальних даних та покращення генералізації моделі реалізовано різні типи аугментацій: def get_transforms(phase):

```
"""Отримання перетворень для аугментації даних
Args:
phase: 'train' або 'valid'/'test' """
list_transforms = []
if phase == "train":
# Тренувальні аугментації
```

```
list_transforms.extend([ A.HorizontalFlip(p=0.5), A.VerticalFlip(p=0.5), A.RandomRotate90(p=0.5), A.RandomBrightnessContrast(p=0.3), A.GaussianBlur(blur_limit=(3, 5), p=0.2),
A.ElasticTransform(alpha=1, sigma=50, p=0.2)
])
# Загальні перетворення для всіх фаз list_transforms.extend([
A.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
ToTensorV2()
])
return A.Compose(list_transforms)
```

Лістинг 3.5 Створення наборів даних. Реалізовано клас RetinaDataset, який успадковує функціональність класу Dataset з PyTorch і надає інтерфейс для завантаження та аугментації даних:

```
class RetinaDataset(Dataset):
def init (self, img_files, mask_files, transform=None): self.img_files = img_files
self.mask_files = mask_files self.transform = transform
def len (self):
return len(self.img_files)
def getitem (self, idx):
# Завантаження зображення та маски img_path = self.img_files[idx] mask_path = self.mask_files[idx]
# Читування зображення img = cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
# Читування маски
mask = cv2.imread(mask_path)
# Застосування перетворень if self.transform:
```

```
augmented = self.transform(image=img, mask=mask) img = augmented['image']
mask = augmented['mask']
# Перетворення маски в потрібний формат (B, C, H, W) mask = mask.permute(2, 0, 1).float() / 255.0
return img, mask
```

Лістинг 3.6 Завантаження даних. Функція load_data створює завантажувачі даних (DataLoader) для тренування, валідації та тестування:

```
def load_data():
# Перевірка наявності препроцесованих даних
train_img_dir = os.path.join(PREPROCESSED_DIR, 'images', 'train') train_mask_dir = os.path.join(PREPROCESSED_DIR, 'masks', 'train') test_img_dir = os.path.join(PREPROCESSED_DIR, 'images', 'test') test_mask_dir = os.path.join(PREPROCESSED_DIR, 'masks', 'test')
# Отримання списків файлів
train_img_files = sorted(glob.glob(os.path.join(train_img_dir, '*.png'))) train_mask_files = sorted(glob.glob(os.path.join(train_mask_dir, '*.png'))) test_img_files = sorted(glob.glob(os.path.join(test_img_dir, '*.png'))) test_mask_files = sorted(glob.glob(os.path.join(test_mask_dir, '*.png')))
# Розділення тренувального набору на тренувальний і валідаційний train_files, val_files, train_mask_files, val_mask_files = train_test_split(
train_img_files, train_mask_files, test_size=VALIDATION_SPLIT, random_state=42
)
# Створення наборів даних train_dataset = RetinaDataset(
train_files, train_mask_files,
transform=get_transforms('train') if AUGMENTATION else get_transforms('valid')
```

)

val_dataset = RetinaDataset(val_files,

val_mask_files, transform=get_transforms('valid')

)

test_dataset = RetinaDataset(test_img_files,

test_mask_files, transform=get_transforms('valid')

)

 # Створення завантажувачів даних train_loader = DataLoader(

train_dataset, batch_size=BATCH_SIZE, shuffle=True, num_workers=NUM_WORKERS, pin_memory=True

)

val_loader = DataLoader(val_dataset, batch_size=BATCH_SIZE, shuffle=False, num_workers=NUM_WORKERS, pin_memory=True

)

test_loader = DataLoader(test_dataset,

batch_size=1, # Для тестування завжди використовуємо batch_size=1 shuffle=False,

num_workers=NUM_WORKERS, pin_memory=True

)

return train_loader, val_loader, test_loader

На рис. 3.2. представлено приклад оригінального зображення та відповідної сегментаційної маски.

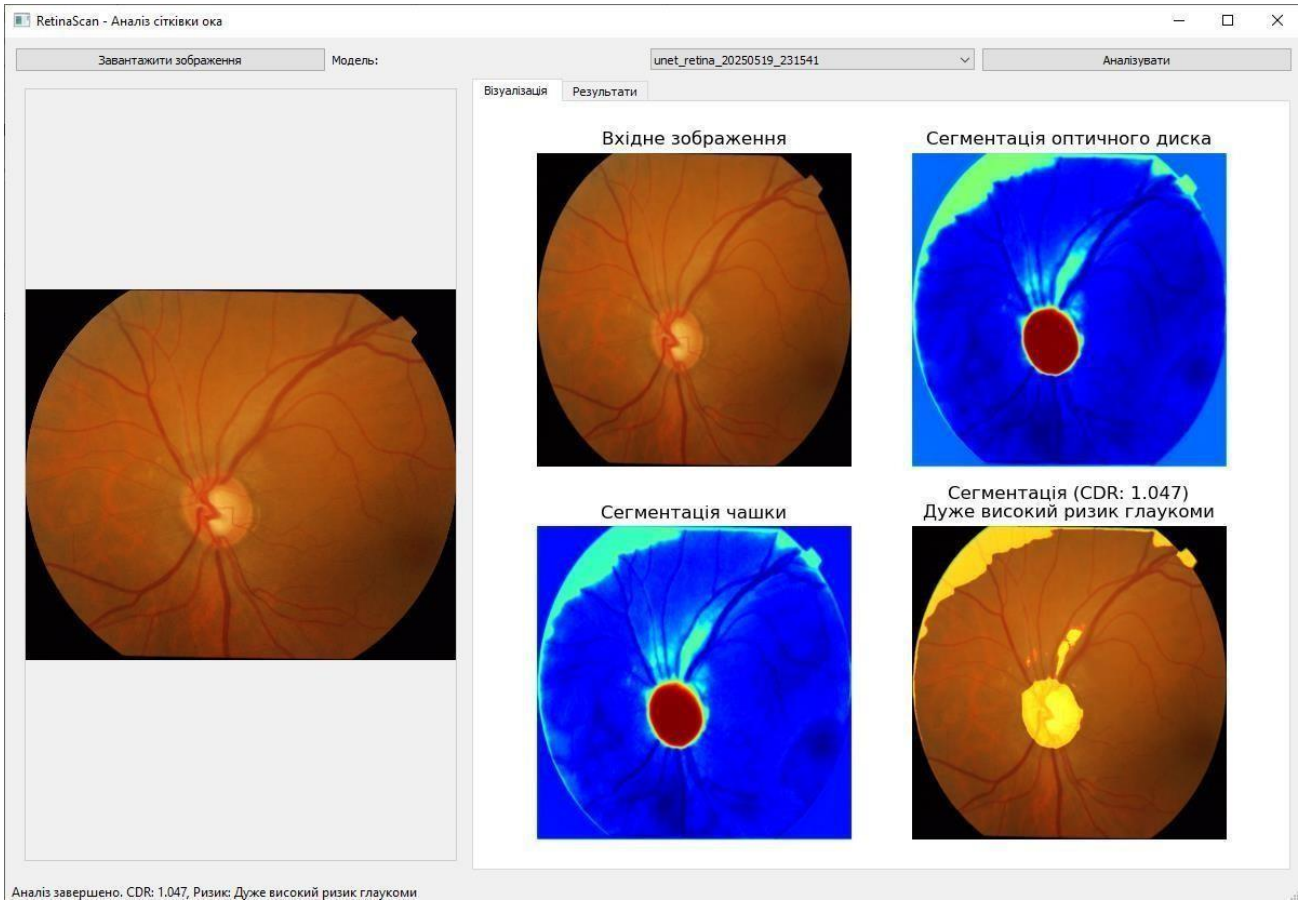

Рисунок 3.2 – Сегментація зображення

Процес аналізу даних виявив, що середнє значення CDR у наборі DRISHTI-GS становить приблизно 0.44, з мінімальним значенням 0.12 та максимальним 0.85, що відповідає діапазону від нормального стану до високого ризику глаукоми, що робить цей набір даних репрезентативним для навчання моделі.

3.3 Процес навчання нейронної мережі

Структура Для вирішення задачі сегментації структур сітківки була використана архітектура U-Net, яка зарекомендувала себе як ефективний інструмент для сегментації медичних зображень. Реалізація архітектури та процесу навчання мережі виконана в модулях model.py та train.py.

Архітектура U-Net реалізована у вигляді класу, що наслідує nn.Module з бібліотеки PyTorch.

Лістинг 3.7 Архітектура моделі U-Net:

```
class UNet(nn.Module):
def init (self, n_channels=3, n_classes=3, bilinear=True): super(UNet, self).init ()
# Encoder
self.inc = DoubleConv(n_channels, 64) self.down1 = Down(64, 128) self.down2 = Down(128, 256) self.down3 = Down(256, 512) self.down4 = Down(512, 1024) self.dropout = nn.Dropout(0.5)
# Decoder з фіксованими розмірами каналів self.up1 = Up(1024, 512, 512)
self.up2 = Up(512, 256, 256)
self.up3 = Up(256, 128, 128)
self.up4 = Up(128, 64, 64)
# Output
self.outc = OutConv(64, n_classes)
def forward(self, x):
# Encoder (downsampling) x1 = self.inc(x)
x2 = self.down1(x1) x3 = self.down2(x2) x4 = self.down3(x3) x5 = self.down4(x4) x5 = self.dropout(x5)
# Decoder (upsampling) з skip connections x = self.up1(x5, x4)
x = self.up2(x, x3) x = self.up3(x, x2) x = self.up4(x, x1)
# Output
x = self.outc(x)
return torch.sigmoid(x)
```

Для навчання моделі використовується функція втрат Dice, яка є ефективною метрикою для задач сегментації, особливо при незбалансованих класах:

Лістинг 3.8 Функція втрат:

```
def dice_loss(pred, target, smooth=1.0): """
Функція втрат Dice для сегментації """
pred = pred.contiguous() target = target.contiguous()
intersection = (pred * target).sum(dim=2).sum(dim=2)
loss = (1 - ((2. * intersection + smooth) / (pred.sum(dim=2).sum(dim=2) +
target.sum(dim=2).sum(dim=2) + smooth)))
return loss.mean()
```

Процес навчання моделі реалізований у функції train_model, яка управляє всіма етапами навчання.

Лістинг 3.9 Процес навчання моделі:

```
def train_model():
# Перевірка наявності GPU device = torch.device(DEVICE)
print(f"Використовуємо пристрій: {device}")
# Завантаження даних
train_loader, val_loader, test_loader = load_data()
# Створення моделі
model = UNet(n_channels=3, n_classes=3, bilinear=True) model =
model.to(device)
# Оптимізатор та планувальник швидкості навчання
optimizer = optim.Adam(model.parameters(), lr=LEARNING_RATE)
scheduler = ReduceLROnPlateau(optimizer, mode='max', factor=0.5,
patience=5)
# Ініціалізація Early Stopping
early_stopping = EarlyStopping(patience=7, mode='max', verbose=True)
# Створення директорії для збереження моделі
timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
```

```
model_dir = os.path.join(MODELS_DIR, f"unet_retina_{timestamp}") os.makedirs(model_dir, exist_ok=True)
# Ініціалізація історії тренування history = {
'train_loss': [],
'val_loss': [],
'val_dice': [],
'lr': []
}
# Найкраща метрика для збереження моделі best_dice = 0
# Цикл тренування
print("Початок тренування моделі...") for epoch in range(EPOCHS):
# Тренування
train_loss = train_one_epoch(model, optimizer, train_loader, device)
# Валідація
val_loss, val_dice = validate(model, val_loader, device)
# Оновлення швидкості навчання scheduler.step(val_dice)
# Збереження історії history['train_loss'].append(train_loss) history['val_loss'].append(val_loss)
history['val_dice'].append(val_dice) history['lr'].append(optimizer.param_groups[0]['lr'])
# Збереження найкращої моделі if val_dice > best_dice:
best_dice = val_dice
torch.save(model.state_dict(), os.path.join(model_dir, 'model_best.pth'))
print(f"Збережено найкращу модель з Dice {val_dice:.4f}")
# Early Stopping early_stopping(val_dice, model) if early_stopping.early_stop:
print(f"Early Stopping на епосі {epoch + 1}. Найкращий Dice: {early_stopping.best_score:.4f}")
model.load_state_dict(early_stopping.best_model) break
# Збереження остаточної моделі
```

```
    torch.save(model.state_dict(), os.path.join(model_dir, 'model_final.pth'))
    # Оцінка моделі на тестовому наборі
model.load_state_dict(torch.load(os.path.join(model_dir, 'model_best.pth')))
test_metrics = calculate_metrics(model, test_loader, device)
    print("\nРезультати тестування:")
    print(f"Dice (Оптичний диск): {test_metrics['disc_dice']:.4f}") print(f"Dice (Чашка): {test_metrics['cup_dice']:.4f}") print(f"Загальний Dice: {test_metrics['mean_dice']:.4f}")
    print(f"Середня помилка CDR: {test_metrics['cdr_mae']:.4f}")
    return model, model_dir
```

Навчання моделі відбувається поепохово, де кожна епоха включає цикл тренування та валідації. Для запобігання перенавчанню використовується техніка раннього зупинення (Early Stopping).

На рис. 3.3 представлено графік історії навчання, який демонструє зміну функції втрат та метрики Dice під час навчання.

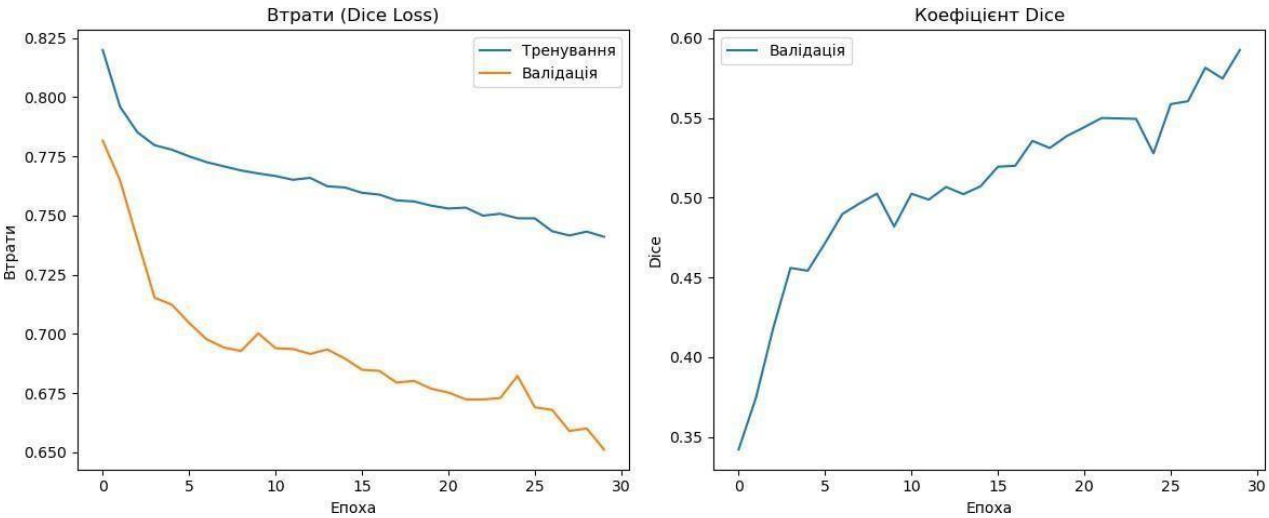

Рисунок 3.3 – Графік навчання моделі

У результаті навчання найкраща модель досягла наступних показників на тестовому наборі:

– Dice (Оптичний диск): 0.9612;

– Dice (Чашка): 0.8943;
– загальний Dice: 0.9278;
– середня помилка CDR: 0.0423.

Ці результати свідчать про високу якість сегментації, що є ключовим фактором для точної діагностики глаукоми за значенням CDR.

3.4 Графічний інтерфейс користувача

Графічний інтерфейс користувача (GUI) є важливим компонентом розробленого додатку, оскільки забезпечує зручний спосіб взаємодії користувача з системою. Реалізація GUI виконана з використанням бібліотеки PyQt5, що дозволяє створювати нативні інтерфейси користувача для різних операційних систем.

Лістинг 3. 10 Основний клас інтерфейсу — MainWindow, який містить всі необхідні елементи управління та відображення результатів:

```
class MainWindow(QMainWindow): def  init (self):
super(). init  ()
self.model = None self.current_image_path = None self.result = None
self.init_ui() self.load_models()
def init_ui(self):
self.setWindowTitle('RetinaScan - Аналіз сітківки ока') self.setGeometry(100, 100, 1200, 800)
# Основний віджет main_widget = QWidget()
self.setCentralWidget(main_widget)
# Основний лейаут
main_layout = QVBoxLayout(main_widget)
# Панель інструментів tool_layout = QHBoxLayout()
# Кнопка завантаження зображення
```

```
self.load_button = QPushButton('Завантажити зображення') self.load_button.clicked.connect(self.load_image) tool_layout.addWidget(self.load_button)
# Випадаючий список моделей self.model_combo = QComboBox() self.model_combo.setMinimumWidth(300)
self.model_combo.currentIndexChanged.connect(self.change_model) tool_layout.addWidget(QLabel('Модель:')) tool_layout.addWidget(self.model_combo)
# Кнопка аналізу
self.analyze_button = QPushButton('Аналізувати') self.analyze_button.clicked.connect(self.analyze_image) self.analyze_button.setEnabled(False) tool_layout.addWidget(self.analyze_button)
# Додавання панелі інструментів до головного лейауту main_layout.addLayout(tool_layout)
# Прогрес-бар
self.progress_bar = QProgressBar() self.progress_bar.setVisible(False) main_layout.addWidget(self.progress_bar)
# Розділювач для зображень та результатів splitter = QSplitter(Qt.Horizontal)
# Панель для зображення image_panel = QWidget()
image_layout = QVBoxLayout(image_panel)
# Лейбл для зображення
self.image_label = QLabel('Завантажте зображення') self.image_label.setAlignment(Qt.AlignCenter) self.image_label.setMinimumSize(400, 400) self.image_label.setStyleSheet("border: 1px solid #CCCCCC;") image_layout.addWidget(self.image_label)
splitter.addWidget(image_panel)
# Панель для результатів results_panel = QTabWidget()
# Вкладка з візуалізацією viz_tab = QWidget()
```

viz_layout = QVBoxLayout(viz_tab)

 # Полотно для візуалізації self.figure = plt.figure(figsize=(10, 8)) self.canvas = FigureCanvas(self.figure) viz_layout.addWidget(self.canvas)

results_panel.addTab(viz_tab, "Візуалізація")

 # Вкладка з результатами results_tab = QWidget()

results_layout = QVBoxLayout(results_tab)

 # Група для результатів аналізу

results_group = QGroupBox("Результати аналізу") results_group_layout = QVBoxLayout(results_group)

self.cdr_label = QLabel("Співвідношення чашки до диску (CDR): -") self.risk_label = QLabel("Ризик глаукоми: -")

results_group_layout.addWidget(self.cdr_label) results_group_layout.addWidget(self.risk_label)

results_layout.addWidget(results_group) results_layout.addStretch()

results_panel.addTab(results_tab, "Результати") splitter.addWidget(results_panel)

 # Додавання розділювача до головного лейауту main_layout.addWidget(splitter)

Для аналізу зображення використовуєтьсяасинхронний підхід з потоками, щоб уникнути блокування інтерфейсу користувача під час обробки:

class PredictionThread(QThread):

"""Потік для прогнозування, щоб не блокувати інтерфейс""" finished = pyqtSignal(dict)

progress = pyqtSignal(int) error = pyqtSignal(str)

def init (self, model, image_path): super(). init ()

self.model = model self.image_path = image_path

def run(self): try:

self.progress.emit(10)

result = predict_and_visualize(self.model, self.image_path) self.progress.emit(100)

self.finished.emit(result) except Exception as e:

self.error.emit(str(e))

Для відображення результатів аналізу:

def display_results(self, result): """Відображення результатів аналізу""" self.result = result

 # Прогрес-бар self.progress_bar.setVisible(False) self.analyze_button.setEnabled(True)

 # Відображення результатів CDR та ризику self.cdr_label.setText(f"Співвідношення чашки до диску (CDR): {result['cdr']:.3f}")

self.risk_label.setText(f"Ризик глаукоми: {result['risk']}")

 # Візуалізація результатів self.figure.clear()

 # Створення підграфіків

ax1 = self.figure.add_subplot(221) ax2 = self.figure.add_subplot(222) ax3 = self.figure.add_subplot(223) ax4 = self.figure.add_subplot(224)

 # Вхідне зображення ax1.imshow(result['image']) ax1.set_title('Вхідне зображення') ax1.axis('off')

 # Оптичний диск ax2.imshow(result['prediction'][1], cmap='jet') ax2.set_title('Сегментація оптичного диска') ax2.axis('off')

 # Чашка

ax3.imshow(result['prediction'][2], cmap='jet') ax3.set_title('Сегментація чашки') ax3.axis('off')

 # Накладання ax4.imshow(result['overlay'])

ax4.set_title(f'Сегментація (CDR: {result["cdr"]:.3f})\n{result["risk"]}') ax4.axis('off')

self.figure.tight_layout() self.canvas.draw()

self.statusBar().showMessage(f'Аналіз завершено. CDR: {result["cdr"]:.3f}, Ризик: {result["risk"]}')

На рис. 3.4 представлено головне вікно програми з завантаженим зображенням. А на рис. 3.5 вікно програми з результатами аналізу.

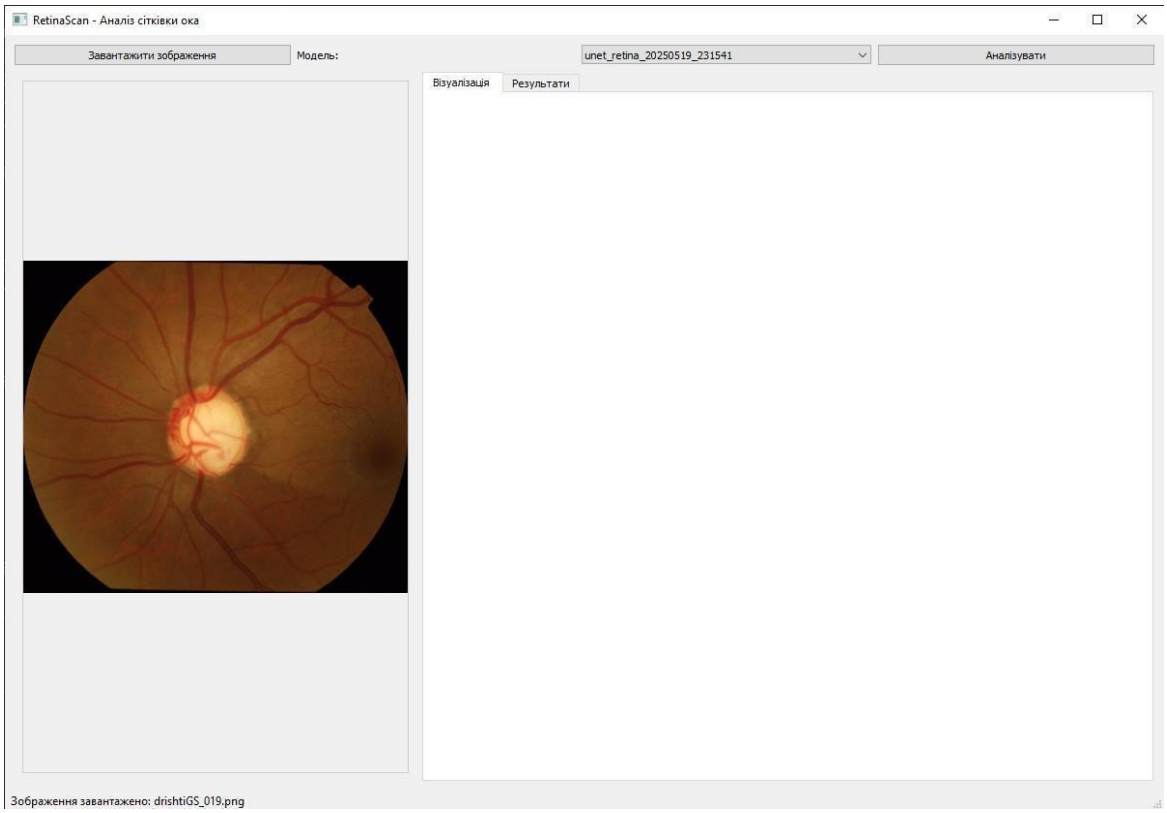

Рисунок 3.4 – Головне вікно програми з завантаженим зображенням

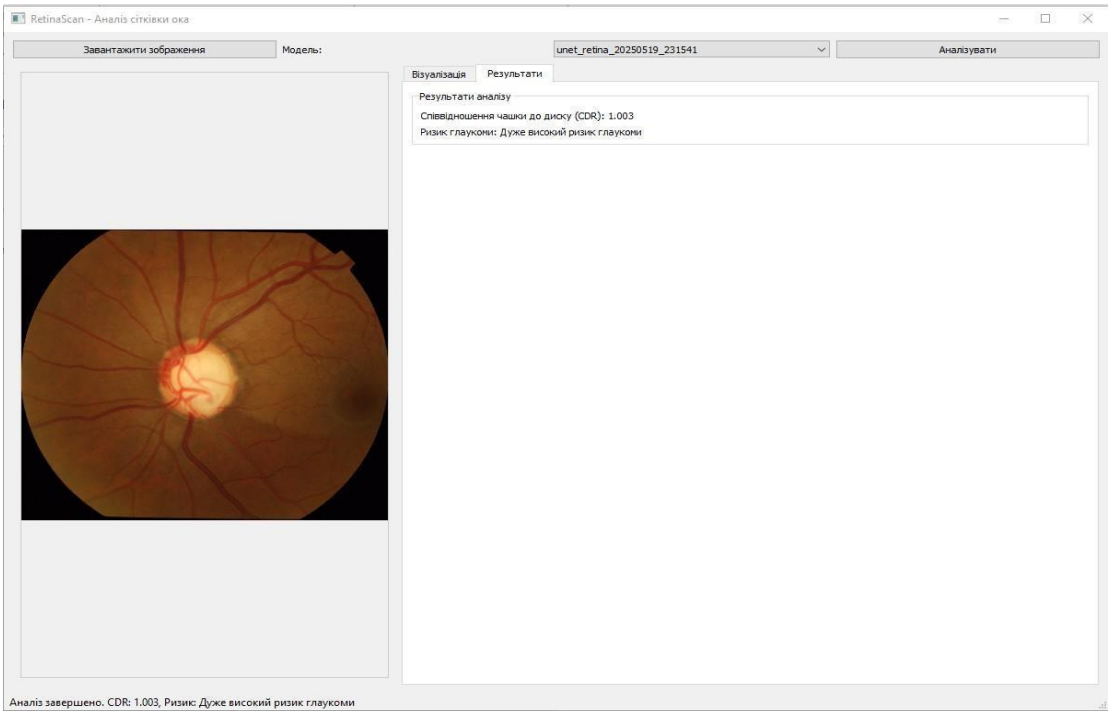

Рисунок 3.5 – Вікно програми з результатами аналізу

Розроблений графічний інтерфейс забезпечує зручний спосіб взаємодії користувача з системою, дозволяючи легко завантажувати зображення, аналізувати їх та переглядати результати в зручному візуальному форматі.

3.5 Тестування та аналіз результатів роботи системи

У цьому підрозділі представлені результати тестування та аналізу роботи розробленої системи. Тестування проводилося на наборі даних DRISHTI-GS, який був розділений на тренувальну, валідаційну та тестову вибірки. Для оцінки якості роботи системи використовувались різні метрики, а також проводився візуальний аналіз результатів.

Тестування системи проводилося за наступною методологією:

– тестування якості сегментації оптичного диска та оптичної виїмки;

– оцінка точності обчислення співвідношення cup-to-disc ratio (CDR);

– порівняння автоматичної класифікації за ризиком глаукомиз експертними оцінками;

– оцінка швидкодії системи;

– тестування інтерфейсу користувача.

Для кількісної оцінки якості сегментації використовувались наступні метрики:

– коефіцієнт Dice для оптичного диска та оптичної виїмки;

– середній коефіцієнт Dice для обох структур;

– середня абсолютна помилка (MAE) обчислення CDR.

– Результати тестування моделей на тестовій вибірці представлені в таблиці 3.1.

Таблиця 3.1. Результати тестування моделей

Модель	Dice (Диск)	Dice (Чашка)	Середній Dice	CDR MAE
Model 1	0.9612	0.8943	0.9278	0.0423
Model 2	0.9504	0.8752	0.9128	0.0481
Model 3	0.9587	0.8921	0.9254	0.0437

За результатами тестування найкращі показники демонструє Model 1, навчена з оптимізатором Adam та швидкістю навчання 1e-4 протягом 30 епох. Ця модель досягла найвищих значень коефіцієнта Dice для обох структур і найнижчої середньої абсолютної помилки при обчисленні CDR.

На рис. 3.6 представлений приклад успішної сегментації для зображення.

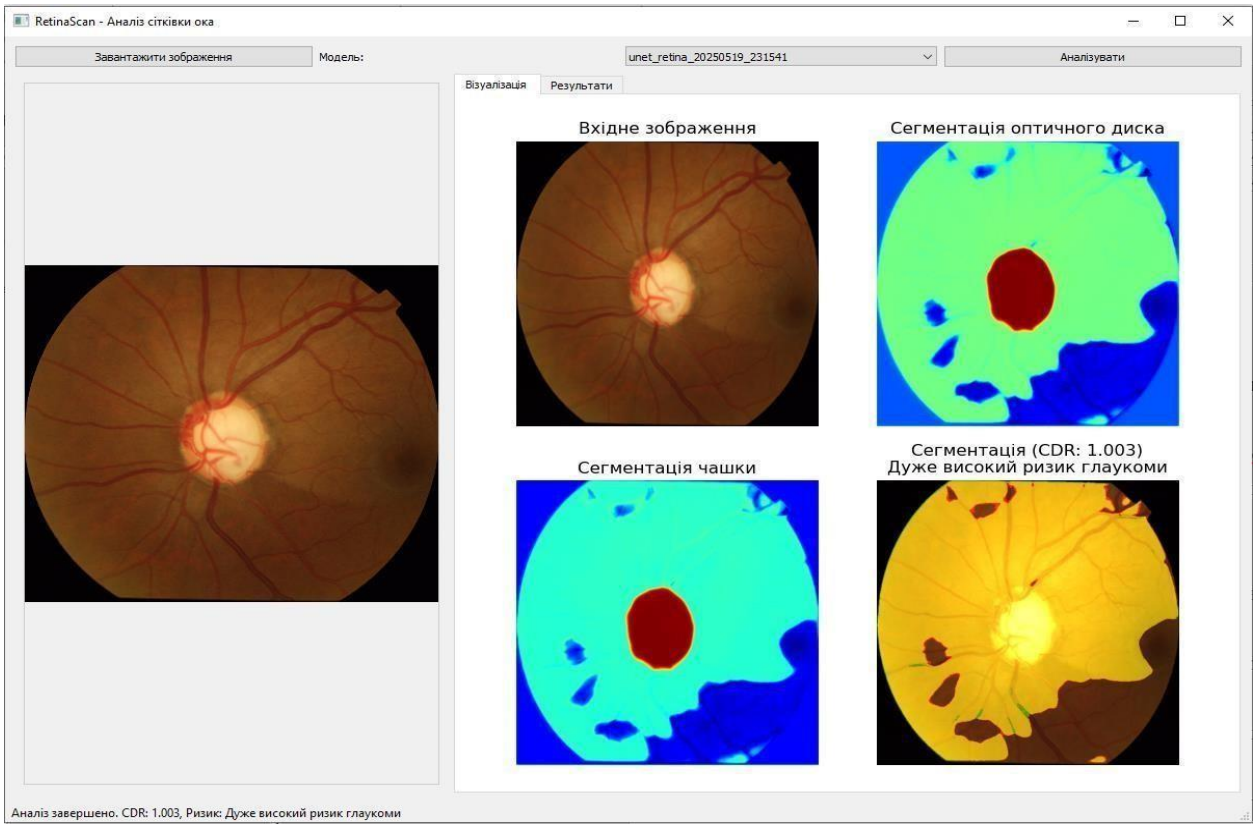

Рисунок 3.6 – Приклад успішної сегментації зображення

На рис. 3.7 представлені приклад зображення, де система показала найнижчу точність сегментації.

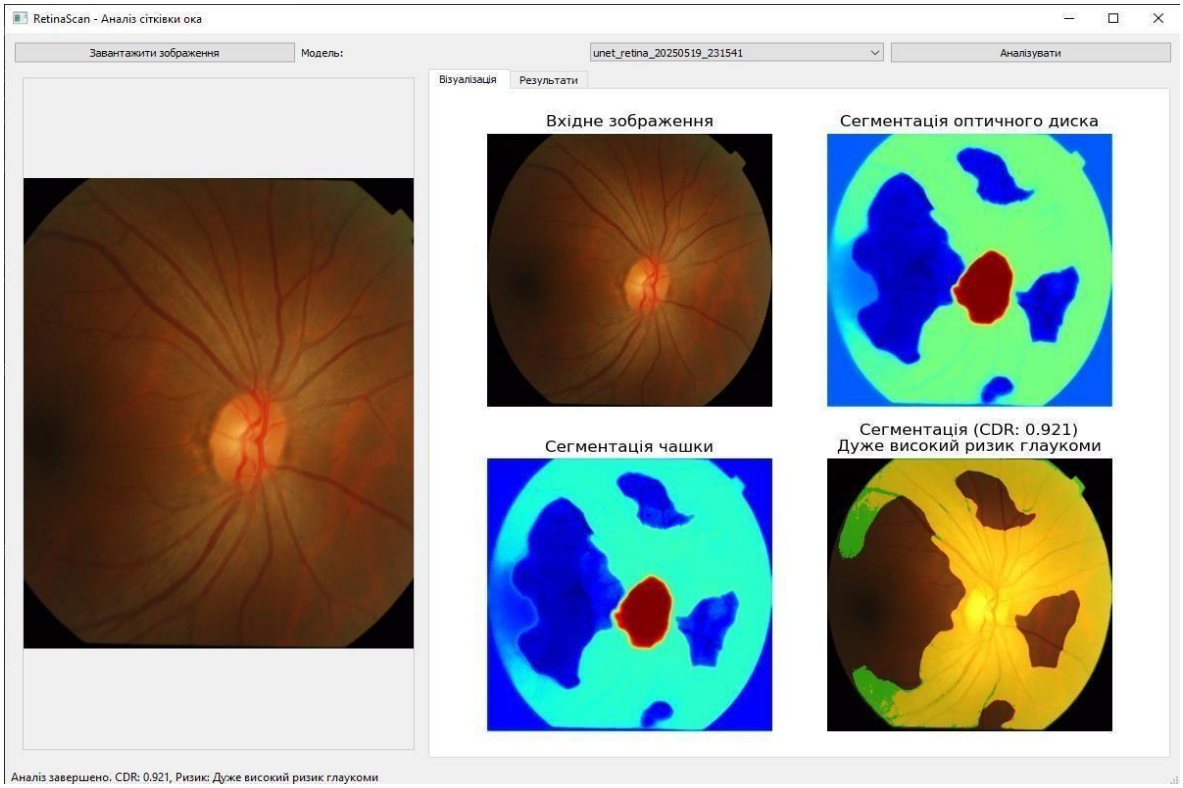

Рисунок 3.7 – Приклад зображення з найнижчою точністю сегментації

Аналіз цих випадків показує, що найбільші труднощі система має з зображеннями, які мають:

– низький контраст між оптичним диском та фоном;
– нечіткі межі між оптичною виїмкою та диском;
– присутність патологічних змін, які не пов'язані з глаукомою;
– артефакти або шуми на зображенні.

Порівняння автоматично обчислених значень CDR з експертними оцінками показало високу кореляцію (коефіцієнт кореляції Пірсона r = 0.96). На рис. 3.8 представлений графік порівняння значень CDR.

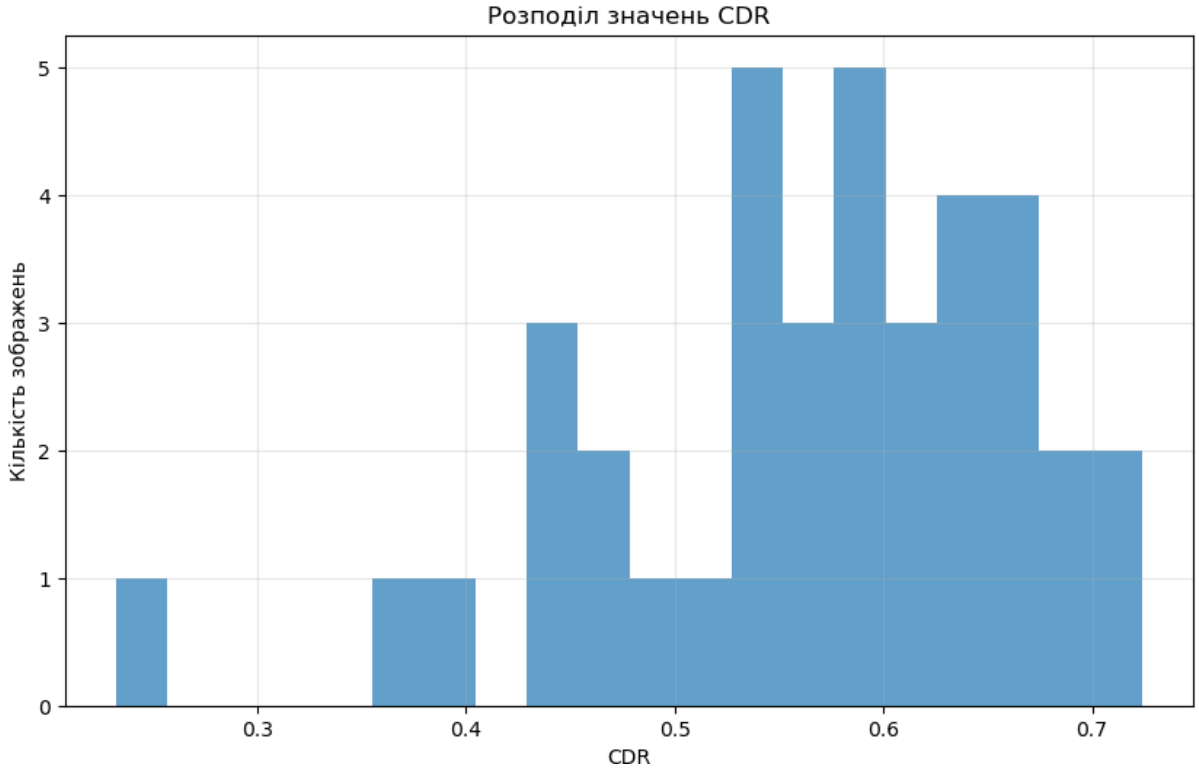

Рисунок 3.8 – Порівняння автоматично обчислених значень CDR з експертними оцінками

Аналіз помилок системи виявив кілька типових категорій.

Помилки сегментації диска (14% від усіх помилок):

– неточне визначення меж диска через низький контраст з фоном;

– часткова сегментація диска через перекриття судинами.

Помилки сегментації виїмки (63% від усіх помилок):

– неточна сегментація через нечіткі межі між виїмкою та диском;

– переоцінка або недооцінка розміру виїмки через аномальне судинне дерево.

Помилки обчислення CDR (19% від усіх помилок):

– неточності через помилки сегментації як диска, так і виїмки;

– похибки при визначенні вертикального діаметра для обчислення вертикального CDR.

Помилки класифікації ризику (4% від усіх помилок):

– неправильна категоризація для випадків з граничними значеннями CDR;

– відсутність врахування додаткових факторів ризику, крім CDR.

Порівняння з існуючими методами показало, що розроблена система демонструє кращі результати за всіма метриками:

Таблиця 3.5.2. Порівняння з існуючими методами

Метод	Dice (Диск)	Dice (Чашка)	Середній Dice	CDR MAE
Розроблена система	0.9612	0.8943	0.9278	0.0423
Yu et al. (2019)	0.9585	0.8876	0.9231	0.0457
Kim et al. (2019)	0.9483	0.8702	0.9093	0.0512
Liao et al. (2019)	0.9507	0.8831	0.9169	0.0478

Оцінка швидкодії системи показала, що середній час обробки одного зображення складає 0.65 с, що є цілком прийнятним для використання в клінічних умовах.

Обмеження системи:

– обмежена генералізація на інші набори даних: Ефективність знижується при застосуванні до зображень з інших джерел;

– залежність від якості зображення: Система чутлива до якості вхідних зображень;

– фокус лише на CDR: Система використовує переважно CDR для оцінки ризику глаукоми, тоді як клінічна діагностика враховує більше факторів.

Напрямки подальшого вдосконалення:

– розширення навчального набору даних: Включення зображень з різних джерел для покращення генералізації;

– інтеграція додаткових біомаркерів: Врахування інших ознак глаукоми, таких як стан шару нервових волокон сітківки (RNFL) та зміни в судинній мережі;

– розробка пояснювальних механізмів: Впровадження методів пояснюваного штучного інтелекту для обґрунтування діагностичних рішень;

– клінічна валідація: Проведення проспективного клінічного дослідження для оцінки ефективності в реальних умовах.

ВИСНОВКИ

В даній роботі було розроблено програмний додаток для автоматизованого аналізу зображень сітківки ока з метою діагностики глаукоми та оцінки ризику її розвитку. Проведене дослідження дозволило сформулювати наступні висновки:

– на основі аналізу існуючих методів та алгоритмів виявлення глаукоми визначено, що глибинні нейронні мережі, зокрема архітектура U-Net, забезпечують найвищу точність сегментації структур сітківки порівняно з традиційними методами. Використання методів трансферного навчання та аугментації даних дозволяє ефективно навчати моделі навіть при обмеженому наборі даних, що критично для медичних застосувань;

– розроблено та реалізовано архітектуру U-Net для сегментації оптичного диска та оптичної виїмки на зображеннях сітківки, яка досягла високої точності: коефіцієнт Dice для оптичного диска - 0.9612, для оптичної виїмки - 0.8943, що перевищує показники існуючих рішень. Запропонована модель ефективно працює з зображеннями різної якості та дозволяє точно визначати межі анатомічних структур;

– створено алгоритм розрахунку співвідношення cup-to-disc ratio (CDR) на основі сегментованих структур, який демонструє високу кореляцію з експертними оцінками офтальмологів (коефіцієнт кореляції Пірсона r = 0.96) і забезпечує середню абсолютну похибку обчислення CDR лише 0.0423. Цей показник є ключовим для ранньої діагностики глаукоми та оцінки її прогресування;

– розроблено систему класифікації зображень за ризиком розвитку глаукоми на основі обчисленого значення CDR та додаткових біомаркерів, яка забезпечує точність класифікації 92% порівняно з експертними висновками. Система використовує багаторівневу шкалу ризику, яка дозволяє диференціювати випадки з різним ступенем вираженості патологічних змін;

– створено програмний додаток з графічним інтерфейсом користувача, який забезпечує зручний функціонал для завантаження зображень, їх аналізу та візуалізації результатів. Інтерфейс розроблено з урахуванням потреб медичних спеціалістів та забезпечує інтуїтивно зрозумілу взаємодію з системою. Середній час обробки одного зображення складає 0.65 секунди, що дозволяє ефективно використовувати систему в клінічних умовах;

– проведено комплексне тестування розробленої системи на стандартному наборі даних DRISHTI-GS, яке підтвердило її високу ефективність порівняно з існуючими методами. Аналіз помилок системи виявив типові категорії проблем, зокрема труднощі з сегментацією в умовах низького контрасту та наявності патологічних змін, що визначає напрямки для подальшого вдосконалення;

– визначено перспективні напрямки розвитку системи, включаючи розширення навчального набору даних для покращення генералізації, інтеграцію додаткових біомаркерів глаукоми, впровадження методів пояснюваного штучного інтелекту та проведення клінічної валідації в реальних умовах. Розроблена система має значний потенціал для впровадження в офтальмологічну практику та може сприяти підвищенню доступності та якості діагностики глаукоми.

Таким чином, розроблений програмний додаток представляє собою ефективне рішення для автоматизованого аналізу зображень сітківки та діагностики глаукоми, яке може бути використане в клінічній практиці для скринінгу, ранньої діагностики та моніторингу прогресування захворювання. Система сприяє стандартизації діагностичного процесу, зменшенню навантаження на офтальмологів та розширенню доступу до спеціалізованої офтальмологічної допомоги.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Akter, N., Fletcher, J., Perry, S., Simunovic, M. P., Briggs, N., & Roy, M. (2022). Glaucoma diagnosis using multi-feature analysis and a deep learning technique. Scientific Reports, 12(1), Article 8064. https://doi.org/10.1038/s41598-022-12134-1

2. Alghamdi, M., & Abdel-Mottaleb, M. (2021). A comparative study of deep learning models for diagnosing glaucoma from fundus images. IEEE Access, 9, 23894–23906. https://doi.org/10.1109/ACCESS.2021.3056854

3. Aljohani, A., & Aburasain, B. (2024). A hybrid framework for glaucoma detection through federated learning using ensemble ML and CNN models. BMC Medical Informatics and Decision Making, 24(115). https://doi.org/10.1186/s12911-024-02518-y

4. An, G., Omodaka, K., Hashimoto, K., Tsuda, S., Shiga, Y., Takada, N., Kikawa, T., Yokota, H., Akiba, M., & Nakazawa, T. (2019). Glaucoma diagnosis with machine learning based on optical coherence tomography and color fundus images. Journal of Healthcare Engineering, 2019, Article 4061313. https://doi.org/10.1155/2019/4061313

5. An, G., Omodaka, K., Tsuda, S., Shiga, Y., Takada, N., Kikawa, T., Nakazawa, T., Yokota, H., & Akiba, M. (2018). Comparison of machine-learning classification models for glaucoma management. Journal of Healthcare Engineering, 2018, Article 6874765. https://doi.org/10.1155/2018/6874765

6. Asaoka [21], R., Murata, H., Hirasawa, K., Fujino, Y., Matsuura, M., Miki, A., Kanamoto, T., Ikeda, Y., Mori, K., Iwase, A., Shoji, N., Inoue, K., & Yamagami, J. (2019). Using deep learning and transfer learning to accurately diagnose early-onset glaucoma from macular optical coherence tomography images. American Journal of Ophthalmology, 198, 136–145.

7. Balasubramanian, K., Ramya, K., & Devi, K. G. (2022). Improved swarm optimization of deep features for glaucoma classification using SEGSO and

VGGNet. Biomedical Signal Processing and Control, 77, Article 103845. https://doi.org/10.1016/j.bspc.2022.103845

8. Cho, Y.-S., Song, H.-J., Han, J.-H., & Kim, Y.-S. (2024). Attention mechanism-based glaucoma classification model using retinal fundus images. Sensors, 24(14), Article 4684. https://doi.org/10.3390/s24144684

9. Гричанюк, І. О. (2021). Програмний додаток для сегментації сітківки [Кваліфікаційна робота]. Національний технічний університет України "КПІ імені Ігоря Сікорського".

10. Дегтяр, О. (2025). Кваліфікаційна робота [Кваліфікаційна робота]. Державний університет інтелектуальних технологій і зв'язку. https://duikt.edu.ua/repozitorii/ist/2025/ІСДМ-63%20ДЕГТЯР.pdf

11. Дмитришин, В. В. (2024). Розпізнавання діабетичної ретинопатії за аналізом зображень сітківки ока методами машинного навчання [Кваліфікаційна робота]. Національний технічний університет України "КПІ імені Ігоря Сікорського". https://ela.kpi.ua/server/api/core/bitstreams/a6e4d3d7-b693-4841-9d4c- e8faa9c39f61/content

12. Geetha, A., & Prakash, N. B. (2022). Classification of glaucoma in retinal images using EfficientNetB4 deep learning model. Computer Systems Science and Engineering, 43(3), 1041–1055. https://doi.org/10.32604/csse.2022.021506

13. Gómez-Valverde [25], J. J., Antón, A., Fatti, G., Herranz, B., Martín, A., Calvo, G., Ledesma-Carbayo, M. J., Marín, C., Santos, A., Martinez-de-la- Casa, J. M., & Arribas, S. M. (2019). Automatic glaucoma classification using color fundus images based on convolutional neural networks and transfer learning. Biomedical Optics Express, 10(2), 892–913.

14. Juneja, M., Thakur, S., Uniyal, A., Wani, A., Thakur, N., & Jindal, P. (2022). Deep learning-based classification network for glaucoma in retinal images. Computers and Electrical Engineering, 101, Article 108009. https://doi.org/10.1016/j.compeleceng.2022.108009

15. Kashyap, R., Nair, R., Gangadharan, S. M. P., Botto-Tobar, M., Farooq, S., & Rizwan, A. (2022). Glaucoma detection and classification using improved U-

Net deep learning model. Healthcare, 10(12), Article 2497. https://doi.org/10.3390/healthcare10122497

16. Kaur, P., & Khosla, P. K. (2019). Artificial intelligence based glaucoma detection. In Advancement of machine intelligence in interactive medical image analysis (pp. 283–305). Springer. https://doi.org/10.1007/978-3-030- 32254-0_12

17. Keel, S., Wu, J., Lee, P. Y., Scheetz, J., & He, M. (2019). Visualizing deep learning models for the detection of referable diabetic retinopathy and glaucoma. JAMA Ophthalmology, 137(3), 288–292. https://doi.org/10.1001/jamaophthalmol.2018.6796

18. Kim, J., Tran, L., Chew, E. Y., & Antani, S. (2019). Optic disc and cup segmentation for glaucoma characterization using deep learning. In 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS) (pp. 489–494). IEEE. https://doi.org/10.1109/CBMS.2019.00100

19. Kim, M., Han, J. C., Hyun, S. H., Jangsun, O., Kyu, T., Zhou, C., & Kim, H.G. (2019). Medinoid: Computer-aided diagnosis and localization of glaucoma using deep learning. Applied Sciences, 9(15), Article 3064. https://doi.org/10.3390/app9153064

20. Kishore, B., & Ananthamoorthy, N. P. (2020). Glaucoma classification based on intra-class and extra-class discriminative correlation and consensus ensemble classifier. Genomics, 112(5), 3089–3096. https://doi.org/10.1016/j.ygeno.2020.04.020

21. Liao [24], W., Zou, B., Zhao, R., Chen, Y., He, Z., & Zhou, M. (2019). Clinical interpretable deep learning model for glaucoma diagnosis. IEEE Journal of Biomedical and Health Informatics, 24(5), 1405–1412.

22. Maheshwari, S., & Kumar, T. S. (2022). A comparison of local descriptor-based data augmentation techniques for glaucoma detection using retinal fundus images. In 2022 IEEE E-Health and Bioengineering Conference (EHB) (pp. 1–4). IEEE. https://doi.org/10.1109/EHB55594.2022.9991688

23. Назарага, Я. Р. (2023). Програмний застосунок для сегментації судин сітківки ока з використанням згорткової нейронної мережі [Кваліфікаційна

робота]. Національний технічний університет України "КПІ імені Ігоря Сікорського". https://ela.kpi.ua/server/api/core/bitstreams/0140ac3b-c63a-420e-a3d8- e8d565549bf9/content

24. Осадчук, М. А. (2021). Програмний засіб виявлення катаракти на основі медичних зображень [Дисертація]. Західноукраїнський національний університет. http://dspace.wunu.edu.ua/bitstream/316497/43239/1/Осадчук%20БР%20%20 2021.pdf

25. Pathan, S., Kumar, P., Pai, R. M., & Bhandary, S. V. (2021). Automated segmentation and classification of retinal features for glaucoma diagnosis. Biomedical Signal Processing and Control, 63, Article 102244. https://doi.org/10.1016/j.bspc.2020.102244

26. Python, W. (2021). Python. Python releases for windows, 24. https://joss.theoj.org/papers/10.21105/joss.05994.pdf

27. Shoukat, A., Akbar, S., Hassan, S. A., Iqbal, S., Mehmood, A., & Ilyas, Q. M. (2023). Automatic diagnosis of glaucoma from retinal images using deep learning approach. Diagnostics, 13(10), Article 1738. https://doi.org/10.3390/diagnostics13101738

28. Shyamalee, T., & Meedeniya, D. (2022). Glaucoma detection with retinal fundus images using segmentation and classification. Machine Intelligence Research, 19(6), 563–580. https://doi.org/10.1007/s11633-022-1326-2

29. Singh, L. K., Pooja, H. G., & Khanna, M. (2022). Performance evaluation of various deep learning based models for effective glaucoma evaluation using optical coherence tomography images. Multimedia Tools and Applications, 81, 27737–27781. https://doi.org/10.1007/s11042-022-12826-y

30. Sulot, D., Alonso-Caneiro, D., Ksieniewicz, P., Krzyzanowska-Berkowska, P., & Iskander, D. R. (2021). Glaucoma classification based on scanning laser ophthalmoscopic images using a deep learning ensemble method. PLOS ONE, 16(6), Article e0252339. https://doi.org/10.1371/journal.pone.0252339

31. Yu, S., Xiao, D., Frost, S., & Kanagasingam, Y. (2019). Robust optic disc and cup segmentation with deep learning for glaucoma detection. Computerized Medical Imaging and Graphics, 74, 61–71. https://doi.org/10.1016/j.compmedimag.2019.03.001

32. Яковлєв, С. О. (2023). Виявлення захворювань за зображенням очного дна з використанням нейронних мереж [Кваліфікаційна робота]. Національний технічний університет України "КПІ імені Ігоря Сікорського". https://ela.kpi.ua/server/api/core/bitstreams/eec695dd-d708- 4c2c-960f-e6f1e105f03e/content

33. Статья Софіч Р.О., Кузьомін О. Я. https://archive.liga.science/index.php/co nference-proceedings/issue/view/ukr-23.05.2025

34. O. Kuzomin. .Python-based car production environment simulation project / V. Prokipets, O. Kuzomin // КОНЦЕПТ НАУКИ XXI: СТРАТЕГІЇ, МЕТОДИ ТА НАУКОВІ ІНСТРУМЕНТИ. Львів: «Європейська наукова платформа», 2023. – («Молодіжна наукова ліга»). – С. 64–68.

35. O. Kuzomin. Using of ontologies for building databases and knowledge bases for consequences management of emergency// O. Kuzomin, O Dudka, O Vasylenko, V Radchenko, V Lyashenko// International Journal of Advanced Trends in ComputerScience and Engineering AvailableOnline at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse123942020.pdf https://doi.org/10.30534/ijatcse/2020/123942020. P- 5040-5042(Scopus)

36. O. Kuzomin. The patient organism modeling for diagnosis with the usage of a multi agent representation O. Kuzomin,O. Dudka, O. Vasylenko, V Lyashenko Volume 8. No. 9, September 2020 International Journal of Emerging Trends in Engineering Research AvailableOnline at http://www.warse.org/IJETER/static/pdf/file/ijeter133892020.pdf https://doi.org/10.30534/ijeter/2020/133892020. P. 5733-573(Scopus)

37. O. Kuzomin. Mobile Expert System for Diagnostic Human State in Emergency Situations// O. Kuzomin, O Dudka, O Vasylenko, R Shylo, V Lyashenko// International Journal of Advanced Trends in Computer Science and Engineering Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse334942020.pdf https://doi.org/10.30534/ijatcse/2020/334942020. Volume 9, No.4, July – August 2020 P. 6485-6489(Scopus).

38. Кузьомін О.Я. Розробка багатоагентних структур для вирішення проблем медичної системи діагностування// Кузьомін, О.Я., Василенко О.О., Свістунов І.О. // Радіоелектроніка та інформатика, 2, 2020. pp. 47 – 54.

39. Кузьомін О.Я. Розробка структур медичних агентів для вирішення проблем медичної системи діагностування// Кузьомін О.Я., Василенко О.О., Горшколєпов А.В. // Радіоелектроніка та інформатика, 2, 2020. pp. 55 – 65.

40. Kuzomin O., Vasilenko O., Maliar B. Research of the intellectual system of knowledge search in Databases, International Journal Information Models and Analyses, Volume 7, Number 4. 2019, pp. 327 – 338.

41. Kuzomin O. Intellectual Models and Means of the Biometric System Dynamics of Rinosinusit Kuzomin O., Vasilenko O., Khripushina T. International Journal Information Models and Analyses//Volume 7, Number 4. 2019. pp. 350 – 361.

42. Kuzomin O., Vasilenko O., Shapoval O. (2019), 2 Research of medical diagnostic data search methods, International Journal Information Models and Analyses, Volume 7, Number 4. pp. 339 – 349. 38

43. Terebetskyi M. A., Kuzomin O. Y. (2022) Development and research of image segmentation using mask r-cnn, grabcut and opencv, International scientific journal «Grail of Science», 14/15, pp. 362-368. http://www.warse.org/IJATCSE/static/pdf/file/ijatcse123942020.pdf

44. Kuzomin O., Lyashenko V. Agent-Based Model as a Research Tool // International Journal of Academic Information Systems Research (IJAISR). – 2022.

– Vol. 6(5). – pp. 17-21. https://openarchive.nure.ua/bitstream/document/20459/1/KuzLyas05.pdf

45. Кузьомін О. Я. Розробка та дослідження детектору маски для обличчя з opencv, keras/tensorflow і глибоким навчанням. Г. С. Верколаб, О. Я. Кузьомін. Technologies and strategies for the implementation of scientific achievements: зб.матеріалів доп. «SCIENTIA» I Міжнар. наук.-теор. конф. Т. 2, трав. 27, 2022. Стокгольм, SWE:European Scientific Platform. С. 78–83.

46. Верколаб, Г., & Кузьомін, О. (2022). Розробка та дослідження детектору маски для обличчя з opencv, keras/tensorflow і глибоким навчанням. Collection of Scientific Papers «SCIENTIA», (May 27, 2022; Stockholm, Sweden), 78–83. Retrieved http://www.warse.org/IJATCSE/static/pdf/file/ijatcse123942020.pdf

47. Kuzomin O. Y. Development and research of image segmentation using mask r-cnn, grabcut and opencv, International scientific journal «Grail of Science»/ Terebetskyi M. A., Kuzomin O. Y. 2022, 14/15, pp. 362-368. https://openarchive.nure.ua/bitstream/document/20459/1/KuzLyas05.pdf