

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту**

До захисту допущено:

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«__» _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Системи і методи штучного
інтелекту»**

спеціальності 122 «Комп'ютерні науки»

**на тему: «Прогнозування діабетичної ретинопатії на основі біомедичних
даних з використанням методів машинного навчання»**

Виконала:

студентка IV курсу, групи КІ-12

Крупко Марія Юріївна _____

Керівник:

доцент кафедри ШІ, к.ф.-м.н., доцент

Пишнограєв І.О. _____

Консультант з економічного розділу:

доцент кафедри економічної кібернетики, к.е.н., доцент,

Рощина Надія Василівна _____

Консультант з нормоконтролю:

фахівець першої категорії кафедри ШІ, к.т.н., доцент,

Комариста Богдана Миколаївна _____

Рецензент:

доцент кафедри ЕК ФММ, к.ф.-м.н., доцент

Лазаренко І.С. _____

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

Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«15» січня 2025 р.

ЗАВДАННЯ

на дипломну роботу студенту

Крупко Марії Юріївни

1. Тема роботи «Прогнозування діабетичної ретинопатії на основі біомедичних даних з використанням методів машинного навчання», керівник роботи керівник роботи Пишнограєв Іван Олександрович, к.ф.-м.н., доцент, затверджені наказом по НН ІПСА від «26» травня 2025 р. № 1759-С.
2. Термін подання студентом роботи «09» червня 2025 року.
3. Вихідні дані до роботи – набір маркованих зображень сітківки ока
4. Зміст роботи
 1. Аналіз предметної області та постановка задачі
 2. Теоретичні основи використання методів глибокого навчання для класифікації медичних зображень
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація результатів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Надія Василівна, доцент, к. е. н.	01.04.2025	31.05.2025

7. Дата видачі завдання «03» лютого 2025 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми ДР	21.04.2025	Виконано
2	Опрацювання наявних джерел щодо діабетичної ретинопатії	28.04.2025	Виконано
3	Підготовка першого та другого розділів	05.05.2025	Виконано
4	Розробка моделей, тренування та тестування	12.05.2025	Виконано
5	Підготовка третього розділу та економічної частини	19.05.2025	Виконано
6	Оформлення розділів відповідно до нормоконтролю	26.05.2025	Виконано
7	Підготовка презентації доповіді	02.06.2025	Виконано
8	Оформлення дипломної роботи	09.06.2025	Виконано

Студент

Марія КРУПКО

Керівник

Іван ПИШНОГРАЄВ

РЕФЕРАТ

Дипломна робота: 106 с., 13 рис., 10 табл., 30 посилань, 1 додаток.

ДІАБЕТИЧНА РЕТИНОПАТІЯ, МЕДИЧНА ДІАГНОСТИКА, ГЛИБИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, НЕЙРОННІ МЕРЕЖІ, КОНВОЛЮЦІЙНА МЕРЕЖА, ЗОБРАЖЕННЯ ОЧНОГО ДНА, МАСКИ СУДИН, БАГАТОКЛАСОВА КЛАСИФІКАЦІЯ.

Об'єктом дослідження є процес автоматизованої діагностики діабетичної ретинопатії за допомогою методів глибокого навчання.

Предметом дослідження є архітектури моделей глибокого навчання та методи обробки зображень для виявлення діабетичної ретинопатії.

Метою роботи є підвищення точності автоматизованої діагностики діабетичної ретинопатії шляхом дослідження та розробки моделей глибокого навчання на основі медичних зображень.

Сучасні системи автоматизованої діагностики діабетичної ретинопатії часто працюють за фіксованими правилами або поверхневими моделями, що не враховують складності візуального представлення хвороби. У цій роботі запропоновано підхід на основі глибокого навчання, який дозволяє ефективно класифікувати зображення очного дна за наявності патологій. Проведено експерименти з різними архітектурами та конфігураціями вхідних даних. На основі отриманих результатів сформовано рекомендації щодо побудови практичної системи підтримки медичної діагностики.

ABSTRACT

Bachelor's thesis: 106 p., 13 figures, 10 tables, 30 references, 1 appendix.

DIABETIC RETINOPATHY, MEDICAL DIAGNOSTICS, DEEP LEARNING, IMAGE CLASSIFICATION, NEURAL NETWORKS, CONVOLUTIONAL NEURAL NETWORK, FUNDUS IMAGES, VESSEL MASKS, MULTICLASS CLASSIFICATION.

The object of this research is the process of automated diagnosis of diabetic retinopathy using deep learning methods and techniques.

The subject of research is the architectures of deep learning models and image preprocessing techniques for diabetic retinopathy diagnosis.

The purpose of the work is to improve the accuracy of diabetic retinopathy detection by developing and evaluating deep learning-based classification models using retinal images.

Modern systems for automated diabetic retinopathy diagnosis often rely on rule-based logic or shallow models, which fail to capture the complex visual patterns of the disease. This work presents a deep learning-based approach for robust classification of fundus images. Multiple architectures and input configurations were evaluated experimentally. Based on the results, practical recommendations were developed for constructing a medical decision support system.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ДІАГНОСТИКИ ДІАБЕТИЧНОЇ РЕТИНОПАТІЇ НА ОСНОВІ БІОМЕДИЧНИХ ДАНИХ	10
1.1 Актуальність теми дослідження	10
1.2 Огляд проблеми діабетичної ретинопатії	11
1.2.1 Медичні аспекти діагностики діабетичної ретинопатії	11
1.2.2 Стадії розвитку хвороби	13
1.3 Аналіз існуючих підходів до прогнозування за допомогою штучного інтелекту	14
1.4 Загальний огляд датасетів та вибір	15
1.4.1 EyePACS Dataset (Kaggle Diabetic Retinopathy Detection Challenge)	16
1.4.2 DDR Dataset (Chinese Diabetic Retinopathy Dataset)	17
1.4.3 Diabetic Retinopathy Dataset (вибраний)	17
1.5 Постановка задачі	18
Висновки до розділу 1	19
РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ МЕТОДІВ ГЛИБОКОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ	21
2.1 Поняття машинного та глибокого навчання	21
2.2 Згорткові нейронні мережі: принцип роботи та архітектура	23
2.3 Ключові компоненти функціонування моделей глибокого навчання	25
2.3.1 Активаційні функції	25
2.3.2 Функції втрат	27
2.3.3 Метрики як інструмент оцінки ефективності моделей	28
2.4 Особливості роботи з медичними зображеннями в задачах глибокого	

	7
навчання	28
2.4.1 Висока роздільна здатність та об'ємність медичних зображень	28
2.4.2 Шуми та артефакти у медичних зображеннях	30
2.4.3 Обмеженість анотованих даних та методи подолання	31
2.4.4 Етичні та нормативні аспекти застосування AI у медичних зображеннях	33
2.5 Проблема дисбалансу класів у медичних датасетах: методи вирішення	36
2.6 Підходи до попередньої обробки зображень в офтальмології	38
Висновки до розділу 2	40
РОЗДІЛ 3 РОЗРОБКА ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МОДЕЛЕЙ ВИЯВЛЕННЯ ДІАБЕТИЧНОЇ РЕТИНОПАТІЇ	41
3.1 Загальний підхід до побудови моделей	41
3.2 Багатокласова модель класифікації	42
3.3 Бінарна модель Healthy vs DR	45
3.3.1. Реалізація та структура моделі	45
3.3.2. Результати	46
3.4 Бінарна модель Mild DR vs решта (рання діагностика)	47
3.4.1. Мотивація: фокус на першу стадію як критичну для раннього втручання	47
3.4.2. Реалізація з використанням EfficientNetB0	48
3.4.3. Результати	49
3.5 Дослідження з використанням масок судин	54
3.5.1. Попередня обробка зображень	54
3.5.2. Побудова окремої моделі на основі зображень масок	55
3.5.3. Результати	56
3.6 Порівняльний аналіз побудованих моделей	59
3.6.1 Огляд моделей та метрик	59

	8
3.6.2 Аналіз результатів	60
3.6.3 Висновки з порівняльного аналізу	61
3.6.4 Пропозиції щодо подальшої роботи	63
Висновки до розділу 3	64
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	64
4.1 Постановка задачі проектування	65
4.2 Обґрунтування функцій програмного продукту	66
4.3 Обґрунтування системи параметрів програмного продукту	69
4.4 Аналіз експертного оцінювання параметрів	73
4.5 Аналіз рівня якості варіантів реалізації функцій	78
4.6 Економічний аналіз варіантів розробки програмного продукту	79
4.7 Вибір кращого варіанту ПП техніко-економічного рівня	85
Висновки до розділу 4	86
ВИСНОВКИ	88
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	94

ВСТУП

Діабетична ретинопатія (ДР) є однією з провідних причин втрати зору у працездатного населення. Ускладнення, які спричинені цукровим діабетом, здатні непомітно вражати сітківку ока, і виявлення хвороби на ранніх етапах залишається ключовим фактором у запобіганні тяжким наслідкам. Водночас традиційні методи діагностики потребують значних ресурсів, часу та участі досвідчених офтальмологів.

Сучасний розвиток технологій машинного навчання відкриває нові можливості для автоматизованого аналізу медичних зображень. Зокрема, у задачах офтальмології дедалі частіше застосовуються алгоритми глибокого навчання для виявлення та класифікації ознак патологій на знімках очного дна. Такий підхід дає змогу підвищити точність діагностики, а також масштабувати її для широкого клінічного використання.

Ця робота присвячена дослідженню методів автоматизованої класифікації діабетичної ретинопатії за зображеннями очного дна. Головним є застосування технологій комп'ютерного зору, особливості попередньої обробки зображень, а також побудова моделей, здатних ефективно диференціювати стадії хвороби.

Тут і нижче використані такий інструмент штучного інтелекту як чат-бот з генеративним штучним інтелектом ChatGPT, виключно для корегування та редагування тексту, створеного автором цієї дипломної роботи, на основі автоматизованої перевірки граматики, структури та стилю, що відповідає Політиці використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського (протокол №11 Вченої ради КПІ ім. Ігоря Сікорського від 11 грудня 2023 р.).

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ДІАГНОСТИКИ ДІАБЕТИЧНОЇ РЕТИНОПАТІЇ НА ОСНОВІ БІОМЕДИЧНИХ ДАНИХ

1.1 Актуальність теми дослідження

Дослідивши тему діабетичної ретинопатії, стало зрозуміло, що це одна з найпідступніших форм вторинного прояву захворювання цукрового діабету. Вона часто залишається непоміченою до моменту, коли повернути зір уже неможливо.

На відміну від болю чи порушення функцій, які змушують людину одразу звернутися по медичну допомогу, діабетична ретинопатія може прогресувати роками абсолютно безсимптомно. За даними Американської академії офтальмології, понад 80% людей з діабетом, які живуть з ним понад 20 років, мають ознаки ретинопатії, і при цьому в більшості з них навіть не було жодних змін зору на ранніх стадіях [1]. Більше того, у багатьох пацієнтів, особливо у країнах із низьким доступом до медицини, саме втрата зору є першим сигналом, що з їхнім організмом щось не так.

Це ускладнення вражає сітківку – найделікатнішу і найчутливішу частину зорового апарату, яка відповідає за сприйняття світла, кольору та руху. Її клітини не регенеруються, тож кожні, навіть мікроскопічні пошкодження, можуть мати незворотні наслідки. Довготривала гіперглікемія пошкоджує капіляри, що живлять сітківку, зменшуючи притік кисню. У відповідь організм починає утворювати нові судини, які мають порушену структуру і є крихкими, це часто лише погіршує ситуацію [2].

Ці зміни не відбуваються миттєво. Протягом тривалого періоду людину може нічого не турбувати. І тому своєчасне виявлення патологічних процесів у сітківці є критичним.

У дослідженні UKPDS (UK Prospective Diabetes Study), яке триває вже понад 40 років, було показано, що саме судинні ураження сітківки є важливим предиктором загального мікросудинного стану людини з діабетом. Тобто, коли

видно зміни в очному дні, це свідчить, що патологічні процеси вже охоплюють і серце, і нирки, і мозок [4].

Ще один тривожний момент – це зміщення вікової межі. Якщо раніше діабетична ретинопатія асоціювалася в першу чергу з людьми старшого віку, то зараз, через поширення цукрового діабету II типу серед молоді, ці ураження з'являються вже у 25-30 років. Це створює величезний виклик для системи охорони здоров'я, бо мова йде як про якість життя, так і про втрату працездатності, інвалідизацію та психологічний стрес.

Насправді, діабетична ретинопатія – це маркер загального стану організму. Саме тому сучасним науковцям, відкривається перспектива для розробки інтелектуальних моделей прогнозування, які за даними зображення сітківки здатні діагностувати захворювання і оцінити ризики ураження інших органів.

1.2 Огляд проблеми діабетичної ретинопатії

1.2.1 Медичні аспекти діагностики діабетичної ретинопатії

Діабетична ретинопатія – це ускладнення, яке виникає внаслідок тривалого впливу підвищеного рівня цукру в крові на судини сітківки ока [5]. Сама по собі сітківка є найчутливішою частиною ока, щось на зразок біологічної матриці, яка перетворює світло на електричні сигнали, які передаються в мозок.

Але коли діабет триває роками, він поступово порушує функцію дрібних судин. Судинні стінки стають тоншими, у деяких місцях з'являються мікрокрововиливи, а тканини навколо починають страждати від браку кисню. У відповідь на це організм намагається допомогти тим, що створює нові судини як

зображено на рис. 1.1., але ці нові капіляри надто тендітні та часто лише погіршують ситуацію.

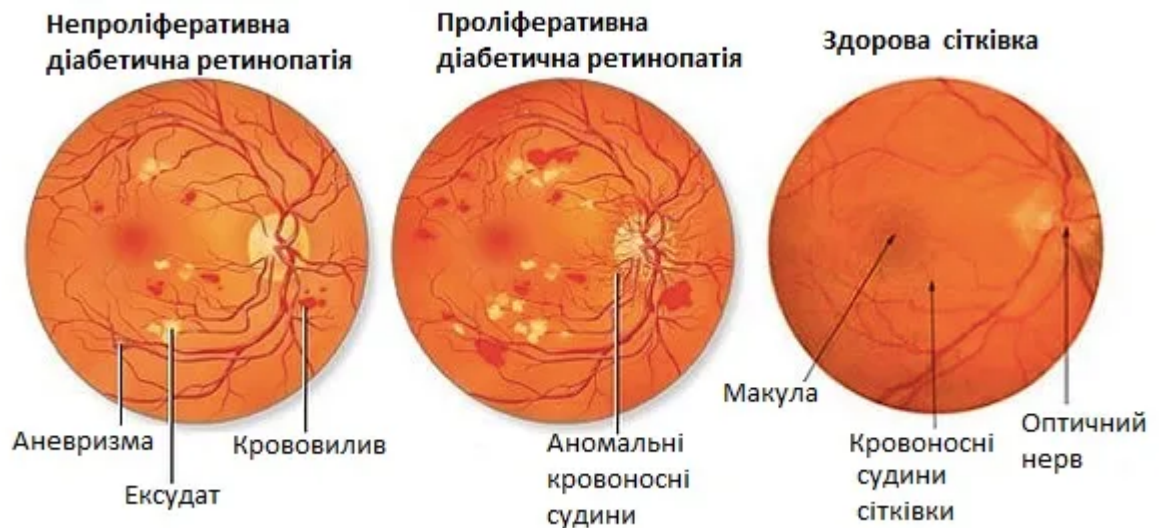


Рисунок 1.1 – Діабетична ретинопатія: ураження сітківки ока внаслідок тривалого перебігу цукрового діабету [6]

Процес цей не відбувається раптово, на ранніх стадіях людина зазвичай не відчуває змін. Саме тому діагностика діабетичної ретинопатії має критичне значення: найнебезпечніші ушкодження – це ті, які розвиваються непомітно. Звідси і важливість регулярного офтальмологічного скринінгу, особливо для людей із тривалим стажем діабету або тих, хто має проблеми з контролем рівня глюкози [7].

Класичний підхід до діагностики включає візуальний огляд очного дна за допомогою спеціального обладнання (наприклад, офтальмоскопа або щілинної лампи). Проте в сучасних клініках усе частіше використовують цифрові методи: отримують фотографії сітківки у високій роздільній здатності, які можна проаналізувати як лікарем, так і автоматизованою системою. На цьому моменті, замість суб'єктивного людського аналізу зображення може бути використаний алгоритм, який знаходить дрібні ознаки ушкодження ще до того, як їх встигне

помітити лікар [8].

1.2.2 Стадії розвитку хвороби

Залежно від морфологічних змін у сітківці, ДР поділяється на кілька стадій, що мають різні клінічні прояви та ризики для зору. Класифікація, яка була запропонована дослідниками Early Treatment Diabetic Retinopathy Study (ETDRS), є загально визнаною та широко використовується в клінічній практиці [9]. Розглянемо стадії діабетичної ретинопатії.

1. М'яка непроліферативна діабетична ретинопатія (NPDR): характеризується наявністю мікроаневризм – невеликих випинань стінок капілярів, які можуть спричиняти незначні крововиливи. Зміни зазвичай не впливають на зір, але є першими ознаками ураження сітківки.
2. Помірна NPDR: відзначається збільшенням кількості мікроаневризм, крововиливів, а також появою твердих ексудатів і ватних плям (cotton wool spots), що свідчать про ішемію сітківки.
3. Важка NPDR: характеризується значними внутрішньосітківковими крововиливами в усіх чотирьох квадрантах, венозною сегментацією та наявністю внутрішньосітківкових мікросудинних аномалій (IRMA). Ці зміни свідчать про високий ризик переходу до проліферативної стадії.
4. Проліферативна діабетична ретинопатія (PDR): найбільш прогресивна стадія, при якій відбувається неоваскуляризація (утворення нових, крихких кровоносних судин, які можуть призводити до крововиливів у склоподібне тіло, фіброзу та відшарування сітківки).

1.3 Аналіз існуючих підходів до прогнозування за допомогою штучного інтелекту

Зі зростанням кількості хворих на цукровий діабет, і відповідно, ускладнень, спричинених ним, проблема раннього виявлення діабетичної ретинопатії набула особливої актуальності. Саме тому, впродовж останніх десяти років, дослідницька спільнота звернула особливу увагу на можливості штучного інтелекту, зокрема комп'ютерного зору та глибокого навчання, для автоматизованої діагностики цієї патології. Вже на сьогодні існують десятки моделей, які побудовані на базі згорткових нейронних мереж (CNN), які досягають результатів, близьких до діагностики офтальмологом із досвідом.

Піонером у сфері застосування глибокого навчання для автоматизованої діагностики діабетичної ретинопатії можна вважати дослідження [10], у якому нейронну мережу архітектури Inception-v3 було застосовано до великого масиву кольорових зображень очного дна. Це дозволило досягти високого рівня точності — площа під кривою (AUC) становила 0,991 у задачі бінарної класифікації «наявність / відсутність діабетичної ретинопатії». Надалі ця модель стала основою для створення комерційного інструменту IDx-DR, який у 2018 році був схвалений FDA (США) як перша автономна система діагностики діабетичної ретинопатії, що функціонує без залучення лікаря [11].

Проте, навіть такі на перший погляд вражаючі результати мають важливе застереження. Більшість систем, які демонструють високу точність, працюють у режимі бінарної класифікації, тобто визначають лише наявність або відсутність ретинопатії. Коли ж завдання ускладнюється до класифікації за стадіями, навіть найсучасніші архітектури стикаються з падінням точності. Так, як показано у змаганні Kaggle Diabetic Retinopathy Detection (2015), навіть найкращі моделі стикалися з труднощами класифікації між суміжними стадіями захворювання, особливо між 1 і 2, або 2 і 3 рівнями за шкалою ICDR (International Clinical Diabetic Retinopathy scale) [12]. Основними причинами цього є як візуальна

подібність симптомів між сусідніми стадіями, так і обмежена кількість якісно розмічених медичних зображень.

Серед сучасних підходів до автоматизованої діагностики діабетичної ретинопатії особливе місце посідають згорткові нейронні мережі (CNN), а також їх модифікації – ResNet, EfficientNet, Vision Transformers (ViT) та гібридні архітектури, які поєднують зображення з клінічними показниками пацієнта. Так, у дослідженні [13] було показано, що інтеграція лабораторних даних (наприклад, рівня HbA1c або тривалості захворювання на діабет) з візуальними ознаками дозволяє підвищити точність класифікації на 7-10%. Такий мультиінформаційний підхід розглядається як перспективний напрям розвитку клінічних систем підтримки прийняття рішень.

Окрему увагу заслуговують роботи, в яких для попередньої обробки даних використовуються методи сегментації судин сітківки. Використання архітектур на кшталт U-Net або DeepVesselNet дозволяє виділяти найбільш інформативні ділянки зображення, що суттєво покращує інтерпретованість моделей та їх здатність ідентифікувати патологічні зміни [14].

Попри всі технічні досягнення, мультикласова класифікація діабетичної ретинопатії залишається складним завданням. Моделі, навчені на одній популяції або одному типі зображень, часто втрачають точність при перенесенні на нові дані, що вказує на проблему узагальнюваності (generalization). Саме тут відкривається простір для нових досліджень, зокрема, для точнішої бінарної класифікації, інтерпретованих моделей та аналізу обмежень існуючих архітектур.

У цій роботі зроблено спробу адаптувати сучасні моделі для вирішення задачі автоматизованого виявлення діабетичної ретинопатії. Було зосереджено увагу на порівнянні ефективності підходів при різних формулюваннях задачі: класифікації "здоровий/хворий", а також розпізнавання стадій.

1.4 Загальний огляд датасетів та вибір

Побудова моделі для прогнозування діабетичної ретинопатії на основі зображень очного дна вимагає якісних, структурованих та відповідним чином анотованих даних. Попри те, що у відкритому доступі існує низка датасетів для цієї задачі, кожен з них має як сильні сторони, так і істотні обмеження. У цьому підрозділі здійснено огляд найбільш популярних відкритих датасетів, а також обґрунтовано вибір основного джерела для дослідження.

1.4.1 EyePACS Dataset (Kaggle Diabetic Retinopathy Detection Challenge)

Найвідомішим і найбільшим набором даних для задачі автоматичної діагностики діабетичної ретинопатії є EyePACS Dataset, який було використано у змаганні Diabetic Retinopathy Detection на Kaggle у 2015 році [15]. Він містить близько 88 000 зображень очного дна, з мітками стадії ретинопатії від 0 до 4. Хоча обсяг вражає, сам набір має низку критичних труднощів для практичного використання:

- зображення дуже неоднорідні за якістю: деякі розмиті, пересвічені або з затемненнями по краях;
- відсутність відповідних масок чи анотацій судин, що ускладнює інтерпретацію моделі.

1.4.2 DDR Dataset (Chinese Diabetic Retinopathy Dataset)

Іншим цікавим, але обмеженим набором є DDR Dataset, зібраний китайською дослідницькою групою [16]. Він містить близько 13 000 зображень, і підтримує багатокласову класифікацію. Проте основним недоліком є те, що дані етнічно та географічно зосереджені на пацієнтах з Китаю, що зменшує узагальнюваність моделей.

1.4.3 Diabetic Retinopathy Dataset (вибраний)

Для цієї дипломної роботи було обрано Diabetic Retinopathy Dataset, опублікований користувачем Sachin Kumar на Kaggle [17]. Хоча цей набір має порівняно невеликий розмір, 3200 зображень, розподілених по 5 класах (від 0 до 4 стадії хвороби), його переваги суттєво переважають недоліки в обсязі:

- якісні кольорові зображення з централізованим оглядом диска зорового нерва і макули, без критичних артефактів;
- простота доступу: формат PNG, уніфіковані назви та анотації, що пришвидшує етап попередньої обробки;
- підходить як для бінарної класифікації (Healthy / DR), так і для повноцінної 5-класової градації.

Крім того, невеликий розмір датасету дозволяє швидко проводити експерименти, особливо в умовах обмежених обчислювальних ресурсів, що є критичним для роботи у середовищі Google Colab. Саме ці властивості зробили датасет оптимальним вибором для цілей цієї роботи.

1.5 Постановка задачі

Метою даної дипломної роботи є розробка програмної системи для автоматизованого визначення стадії діабетичної ретинопатії на основі аналізу зображень очного дна пацієнтів з використанням методів глибокого навчання.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- 1) зібрати та підготувати медичний датасет із класифікованими зображеннями сітківки, з урахуванням балансу між класами;
- 2) реалізувати етапи попередньої обробки зображень, зокрема нормалізацію, масштабування та виділення судинної структури;
- 3) побудувати архітектуру моделі глибокого навчання, здатну проводити багатокласову класифікацію зображень (від здорової сітківки до важкої стадії ретинопатії);
- 4) забезпечити можливість подальшого вдосконалення моделі шляхом використання методів аугментації, трансферного навчання та оптимізації гіперпараметрів;
- 5) провести навчання, тестування та валідацію моделі із подальшою оцінкою її точності, повноти та F1-міри;
- 6) реалізувати зручний інтерфейс або програмний модуль для демонстрації роботи класифікації.

Програмне рішення має бути реалізоване з використанням актуальних інструментів машинного навчання, таких як PyTorch або TensorFlow, з підтримкою прискореної обробки на GPU. З боку користувача очікується мінімальна взаємодія: достатньо буде завантажити зображення, після чого система повинна автоматично провести його аналіз та надати результат у зручній формі.

Передбачено, що алгоритм функціонуватиме в інтерактивному середовищі, з можливістю візуалізації процесу навчання, класифікації та інших діагностичних метрик. Архітектура моделі повинна бути сумісна з сучасними

глибокими згортковими мережами, такими як EfficientNet або ResNet, і водночас дозволяти швидке перенавчання на нових вибірках. Усе це має бути реалізовано з акцентом на збереження узагальнювальної здатності, щоб уникнути перенавчання при роботі з реальними, іноді неоднорідними даними.

Висновки до розділу 1

У першому розділі здійснено аналіз предметної області, який дав змогу окреслити медичну, соціальну та технологічну значущість проблеми діабетичної ретинопатії.

Початково було наголошено на актуальності теми, яка зумовлена зростанням захворюваності на цукровий діабет у світі, а також підступною природою ретинопатії – ускладненням, яке часто не має симптомів на ранніх стадіях, і може призвести до незворотної сліпоті. Було акцентовано, що це ураження є індикатором загального мікросудинного стану організму, а тому раннє його виявлення має критичне значення для всього здоров'я пацієнта.

У підрозділі, присвяченому медичним аспектам, детально розглянуто патофізіологію розвитку ретинопатії: порушення мікроциркуляції сітківки, формування нових, але тендітних судин, що лише погіршують стан тканин. Наголошено на важливості регулярного офтальмологічного скринінгу, зокрема із застосуванням сучасних цифрових методів (ретинографії та автоматизованого аналізу зображень).

Подальший аналіз зосереджено на існуючих підходах до виявлення діабетичної ретинопатії із використанням штучного інтелекту. Було наведено приклади успішних реалізацій згорткових нейронних мереж (CNN), серед яких особливу роль відіграє модель Inception-v3, яка показала вражаючі результати у задачі бінарної класифікації. Однак відзначено, що більш складна задача, багатокласова класифікація за стадіями, залишається викликом навіть для

сучасних моделей, передусім через візуальну подібність сусідніх стадій і обмежену кількість якісних анотованих зображень.

На основі цього сформульовано основну задачу дослідження: розробити та порівняти різні архітектури нейронних мереж у задачі класифікації діабетичної ретинопатії, включаючи як бінарні, так і багатокласові підходи, з використанням оригінальних зображень сітківки та сегментованих масок судин.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ МЕТОДІВ ГЛИБОКОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ

2.1 Поняття машинного та глибокого навчання

Історично машинне навчання (machine learning, ML) виникло як спроба зробити алгоритми більш гнучкими та здатними адаптуватись до нових даних без необхідності вручну задавати кожне правило. У 1950-х роках з'явилися перші роботи Алана Тюрінга, які заклали підвалини ідеї "машинного інтелекту", а вже в 1959 році Артур Семюел уперше вжив термін "machine learning", працюючи над програмою для гри в шашки, яка самостійно вдосконалювала свої ходи шляхом аналізу минулих партій [18]. Протягом наступних десятиліть розвиток був повільним і обмежувався академічними дослідженнями через нестачу обчислювальних потужностей та даних. Проте вже на початку 2000-х, завдяки поширенню Інтернету, зростанню обсягів даних і здешевленню комп'ютерних обчислень, машинне навчання вийшло за межі лабораторій і почало використовуватись у реальних задачах, від пошукових систем до фінансової аналітики [19].

Класичне машинне навчання ґрунтується на побудові моделей, які вивчають закономірності в даних за допомогою алгоритмів на кшталт логістичної регресії, дерева рішень, методів опорних векторів (SVM) або ансамблевих методів на кшталт Random Forest і Gradient Boosting. Ці підходи показують хорошу ефективність на структурованих табличних даних, але значно втрачають у потужності, коли мова йде про високовимірні та слабоформалізовані джерела інформації, наприклад: зображення, відео, звук.

І тут з'являється глибоке навчання (deep learning, DL). Це підгалузь машинного навчання, яка імітує роботу людського мозку за допомогою багаторівневих штучних нейронних мереж. На відміну від класичних методів машинного навчання, глибокі нейронні мережі здатні самостійно формувати

ознаки, необхідні для класифікації, без попереднього ручного конструювання дескрипторів. Наприклад, у класичному підході для аналізу зображень сітківки потрібно було самостійно задавати параметри розмиття, контрасту, колірних діапазонів, а потім будувати модель на основі вручну витягнутих дескрипторів. У глибокому навчанні достатньо надати нейромережі достатньою кількістю зображень, і вона сама навчиться розпізнавати характерні патерни для кожного класу. Це зменшує людський фактор та відкриває шлях до створення по-справжньому автономних діагностичних систем.

Саме тому глибоке навчання стало проривом у медицині. У 2016 році команда дослідників з Google опублікувала знакове дослідження, в якому згортова нейронна мережа досягла рівня офтальмологів-експертів у виявленні діабетичної ретинопатії на основі зображень очного дна (Gulshan et al., *JAMA*, 2016) [20]. Це стало першим масштабним доказом того, що глибокі нейромережі можуть як асистувати лікарям, так і працювати на рівні клінічної точності.

Сьогодні використання методів глибокого навчання в медичній діагностиці охоплює найрізноманітніші напрямки: від виявлення пухлин на КТ-знімках легень до передбачення серцево-судинних захворювань за ЕКГ [21]. Але саме в офтальмології ці підходи продемонстрували найбільшу зрілість, адже знімки сітківки – це стандартизовані, добре структуровані зображення, які чудово підходять для задач комп'ютерного зору.

У цьому контексті задача класифікації стадій діабетичної ретинопатії набуває особливої актуальності. Вона вимагає чіткого диференціювання між п'ятьма ступенями розвитку захворювання, а це є значно складнішим завданням. Глибоке навчання дає змогу моделювати подібні складні категоріальні переходи, відтворюючи контекстні залежності між пікселями, що для людського ока часто залишаються непомітними.

2.2 Згорткові нейронні мережі: принцип роботи та архітектура

Згорткові нейронні мережі (CNN, Convolutional Neural Networks) є основним інструментом у розпізнаванні візуальних структур у задачах комп'ютерного зору [22]. Саме цей тип архітектур заклав основу для застосування глибокого навчання в медицині, зокрема в аналізі зображень очного дна, рентгенології, дерматоскопії та томографії. CNN здатні автоматично виявляти закономірності та ключові ознаки на зображеннях без потреби в ручному описі ознак, що значно відрізняє їх від класичних алгоритмів машинного навчання.

Основною ідеєю згорткової мережі є поетапне вилучення ознак із зображення. Цей процес базується на застосуванні спеціальних фільтрів (ядра згортки), які проходять через зображення і обчислюють нові значення пікселів, відповідно до виявлених локальних патернів. На початкових рівнях мережа зазвичай виявляє базові ознаки, такі як краї, текстури або градієнти. На наступних рівнях дедалі складніші структури: судинні сітки, гіперінтенсивні ділянки, крововиливи, патологічні зміни.

Архітектура CNN зазвичай складається з трьох основних блоків:

- 1) згорткові шари (convolutional layers);
- 2) шари підвибірки (pooling layers);
- 3) повнозв'язні шари (fully connected layers) [23].

На етапі згортки зображення проходить через набір фільтрів. Це невеликі матриці, які сканують його із кроком (stride) і формують карти ознак (feature maps). Ці карти фіксують активації нейронів, які відповідають за присутність певних ознак у локальних ділянках зображення. Кількість фільтрів задається користувачем і є важливим гіперпараметром (збільшення кількості фільтрів дозволяє моделі вивчити ширший спектр патернів і підвищує обчислювальну складність).

Після кожного згорткового шару зазвичай додається шар підвибірки (наприклад, max-pooling), який зменшує розмірність карти ознак. Це дозволяє зменшити кількість параметрів, покращити узагальнення моделі та зробити її менш чутливою до дрібних локальних зсувів. Найчастіше використовується pooling з вікном 2×2 , який бере максимум зі значень у цьому регіоні. Існують також інші варіанти, як-от average pooling, але max-pooling залишається стандартом у більшості задач. Для наочного розуміння принципу роботи max pooling, розглянемо рис. 2.2.

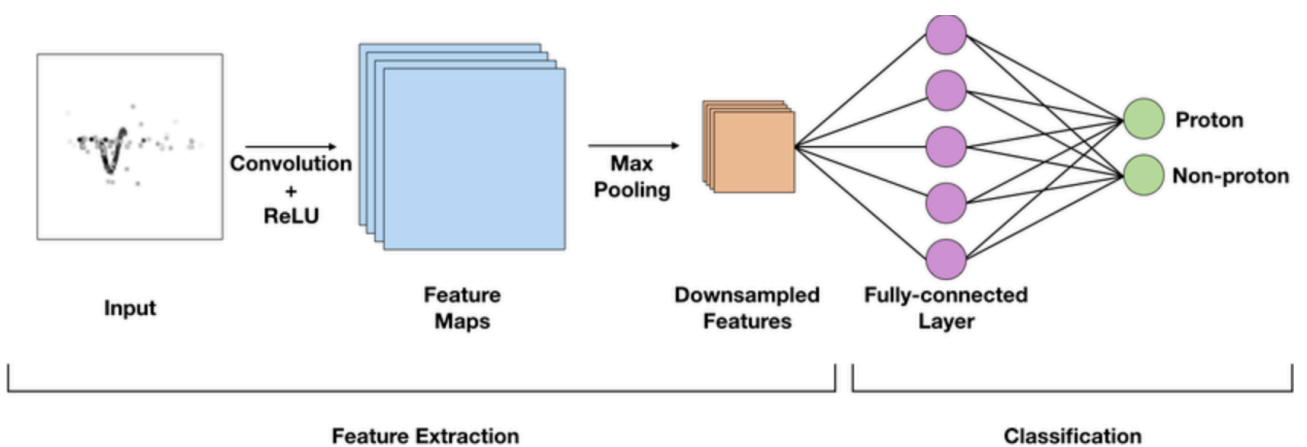


Рисунок 2.2 – Max pooling: зменшення розмірності зображення без втрати ключових ознак

Зображення демонструє, як операція max pooling зменшує розмірність карти ознак, зберігаючи при цьому найважливіші характеристики вхідного зображення.

На фінальному етапі мережа вирівнює ознаки з попередніх шарів за допомогою шару Flatten і передає їх до повнозв'язних шарів. Ці шари виконують класифікацію, опрацьовуючи отримані абстрактні ознаки та формуючи фінальний вихід моделі, наприклад – ймовірності належності до певного класу.

Важливо, що згорткові нейронні мережі навчаються поступово, спочатку зважаються прості фільтри, далі глибші, і на кожному етапі мережа вчиться бачити все більш складні структури. Механізм навчання ґрунтується на мінімізації функції втрат і оновленні ваг за допомогою алгоритмів оптимізації, і ці аспекти розглядаються окремо в наступному підрозділі.

У сучасній практиці CNN можуть бути побудовані як з нуля (тобто з простої послідовності шарів), так і на основі попередньо навчених моделей (наприклад, VGG, ResNet, EfficientNet), які дозволяють досягати високої точності навіть на невеликих медичних вибірках за рахунок переносу знань (transfer learning) [24].

2.3 Ключові компоненти функціонування моделей глибокого навчання

Навчання моделей штучних нейронних мереж базується на складній взаємодії між внутрішніми параметрами мережі та механізмами, які визначають як ці параметри змінюються під час тренування. До основних компонентів, що забезпечують ефективність цього процесу, належать активаційні функції, функції втрат і метрики якості. Кожна з цих складових має критичну роль у досягненні оптимального навчання та коректної генералізації моделі на нові дані.

2.3.1 Активаційні функції

Активаційна функція визначає, яким чином вихід одного нейрона передається далі в мережі. Без неї навіть складна багаторівнева нейронна

мережа працювала б як лінійна модель, що не здатна моделювати складні залежності. Найпоширенішими активаційними функціями є sigmoid, tanh, ReLU (Rectified Linear Unit) та її варіації.

Функція sigmoid перетворює вхідне значення у діапазон від 0 до 1, що надає зручність для задач бінарної класифікації. Математично вона визначається наступним чином:

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

Ця функція має S-подібну форму та забезпечує плавний перехід між вихідними значеннями. Однак вона страждає від проблеми затухаючого градієнта, що ускладнює навчання глибоких мереж [25].

Функція tanh (гіперболічний тангенс), яка також є S-подібною, але має діапазон від -1 до 1 , частково вирішує цю проблему. Вона є масштабованою версією sigmoid-функції:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

Найбільшою популярністю у сучасних глибоких мережах користується функція ReLU (Rectified Linear Unit), яка повертає нуль при від'ємному вході та передає позитивні значення без змін і визначається:

$$f(x) = \max(0, x).$$

Її основна перевага полягає у збереженні незатухаючого градієнта для позитивних значень, що сприяє швидшій та стабільнішій оптимізації [26]. Водночас вона не позбавлена недоліків: наприклад, "мертві нейрони" – ситуація, коли велика кількість нейронів перестає активуватись. Це призвело до

появи модифікацій, таких як Leaky ReLU або Parametric ReLU, які допускають невеликий градієнт і для від'ємних значень.

2.3.2 Функції втрат

Функція втрат (loss function) визначає, наскільки сильно поточні передбачення мережі відрізняються від очікуваних значень. Саме ця функція є основою для обчислення градієнтів і подальшого оновлення ваг у процесі зворотного поширення помилки (backpropagation).

У задачах класифікації найчастіше використовується кросентропія (categorical cross-entropy). Вона визначає відстань між двома розподілами ймовірностей, передбаченим мережею та справжнім (зазвичай у вигляді one-hot-вектора). Формально функція кросентропії для однієї вибірки виглядає так:

$$L = - \sum_{i=1}^c y_i \cdot \log(p_i),$$

де y_i – істинна мітка;

p_i – ймовірність, передбачена мережею для класу i ;

C – кількість класів.

Підхід є ефективним для багатокласової класифікації, так як спрямовує навчання на максимізацію ймовірності правильного класу. У задачах із дисбалансом класів часто застосовують модифікації, зокрема focal loss, яка знижує вплив легких прикладів і фокусується на складних випадках, завдяки додаванню вагової складової та модулюючого множника [27].

2.3.3 Метрики як інструмент оцінки ефективності моделей

Для оцінки продуктивності моделі у класифікаційних задачах використовуються спеціальні метрики, які дозволяють глибше зрозуміти поведінку моделі, особливо в умовах дисбалансу класів. Найбільш поширені метрики – це precision (точність), recall (повнота), F1-score, а також confusion matrix (матриця неточностей).

1. Точність (precision) показує частку правильно класифікованих позитивних прикладів серед усіх, які модель класифікувала як позитивні.
2. Повнота (recall) демонструє, яка частка справжніх позитивних прикладів була виявлена моделлю.
3. F1-score є гармонічним середнім між precision та recall, що робить його особливо корисним у ситуаціях, коли між цими двома показниками існує компроміс.

Матриця неточностей – корисна візуалізація для аналізу класифікаційних помилок. У багатокласовому випадку вона допомагає виявити, які класи найчастіше плутає модель. Розрахунок метрик здійснюється на тестовому наборі, який не використовувався під час тренування, і тоді можна оцінити здатність моделі до узагальнення. У медичних задачах особливу увагу приділяють recall – моделі мають не пропустити критичні випадки захворювання навіть за рахунок зниження точності [28].

2.4 Особливості роботи з медичними зображеннями в задачах глибокого навчання

2.4.1 Висока роздільна здатність та об'ємність медичних зображень

У задачах медичної діагностики, пов'язаних із розпізнаванням патологій за зображеннями сітківки ока, одним із ключових аспектів є висока роздільна здатність вхідних зображень. Це зумовлено самою природою задачі: клінічно значущі ознаки, такі як мікроаневризми, ексудати чи дрібні судинні порушення, мають дуже малий розмір і можуть бути непомітними при недостатній якості зображення. Саме тому більшість відкритих медичних датасетів, призначених для розпізнавання діабетичної ретинопатії, містять зображення розміром у кілька тисяч пікселів по ширині та висоті.

З одного боку, така роздільна здатність дозволяє зберегти максимум візуальної інформації. З іншого, ускладнює безпосередню роботу з цими зображеннями в рамках глибокого навчання. Класичні архітектури згорткових нейронних мереж (CNN), що використовуються в задачах класифікації, зазвичай приймають на вхід зображення розміром 224×224 або 299×299 пікселів. Тому перед обробкою часто застосовується попереднє масштабування або обрізка зображень, що, в свою чергу, потребує обережного підходу: будь-яке зменшення розміру несе ризик втрати критичних діагностичних ознак.

Ще одна складність пов'язана з тим, що на відміну від звичайних зображень, де часто досить працювати з яскравістю або текстурою, у зображеннях очного дна важливу роль відіграє не лише структура, а й колір. Хоча вони є RGB-зображеннями, візуально вони нагадують майже монохромні (із червонувато-коричневим відтінком). В результаті, лише частина каналів може містити справді релевантну інформацію. У наукових працях зустрічаються підходи з перетворенням у колірні простори HSV чи LAB для виділення дрібних деталей, але загалом архітектури глибокого навчання залишаються достатньо гнучкими для роботи і з первинним RGB-представленням.

Крім того, висока роздільна здатність зображень призводить до збільшення обчислювального навантаження. Навіть після зменшення розміру, великі об'єми даних, які характерні для медичних задач, потребують ретельної оптимізації процесу навчання. Найчастіше для цього використовуються методи попередньої обробки (data augmentation, нормалізація піксельних значень, стандартизація), і також пакетна генерація зображень на льоту (наприклад, з використанням генераторів), щоб уникнути перевантаження пам'яті при роботі з великими наборами.

У деяких випадках, коли масштабування неможливе без суттєвої втрати якості, застосовується стратегія розбиття зображення на менші фрагменти (патчі), які аналізуються окремо, а результати об'єднуються на рівні рішення. Однак цей підхід більше притаманний задачам сегментації або локалізації об'єктів. У задачах класифікації, коли кожне зображення відповідає одній мітці, робота з повними кадрами, хоча і потребує більшої обчислювальної потужності, часто залишається пріоритетною.

2.4.2 Шуми та артефакти у медичних зображеннях

У медичній візуалізації якість зображень є критично важливою для точного діагностування та подальшого лікування. Проте, зображення, отримані за допомогою різних методів візуалізації, часто містять шуми та артефакти, які можуть ускладнювати інтерпретацію та аналіз.

Шуми в медичних зображеннях можуть виникати через обмеження обладнання, рух пацієнта під час сканування, низький рівень сигналу або інші фактори. Наприклад, у магнітно-резонансній томографії (МРТ) та комп'ютерній томографії (КТ) шуми можуть призводити до зниження контрастності та чіткості зображень, що ускладнює виявлення патологій.

Артефакти – це небажані структури або спотворення на зображеннях, які не відповідають реальним анатомічним або фізіологічним структурам. Вони можуть виникати через технічні проблеми, такі як несправності детекторів, або фізіологічні фактори, наприклад, рух пацієнта. Артефакти можуть маскувати або імітувати патології, що призводить до помилкових діагнозів.

Застосування методів глибокого навчання для зменшення шумів та артефактів у медичних зображеннях стало перспективним напрямом досліджень. Глибокі нейронні мережі, зокрема згорткові нейронні мережі (CNN), демонструють високу ефективність у задачах фільтрації шумів та відновлення зображень. Наприклад, дослідження показали, що використання CNN дозволяє значно покращити якість зображень КТ при низьких дозах опромінення, зменшуючи шуми без втрати важливої діагностичної інформації.

Проте, тут необхідно зазначити, що надмірне згладжування зображень може призводити до втрати дрібних деталей, важливих для діагностики. Тому важливо досягти балансу між зменшенням шумів та збереженням критичних особливостей зображень. Деякі дослідження також підкреслюють необхідність адаптації моделей глибокого навчання до специфіки медичних зображень, враховуючи їхні унікальні характеристики та вимоги [29].

2.4.3 Обмеженість анотованих даних та методи подолання

Однією з основних складностей при роботі з медичними зображеннями є те, що якісних анотованих даних дуже мало. Розмітка таких даних – це процес, який потребує участі спеціалістів із високою кваліфікацією, наприклад лікарів-офтальмологів чи радіологів. Вони витрачають багато часу на те, щоб правильно позначити кожне зображення, тому зібрати великий набір даних із достовірною розміткою – дуже складно. Крім того, трапляється, що деякі стани

організму, наприклад рідкісні захворювання, зустрічаються вкрай рідко, і через це у датасеті практично немає достатньої кількості прикладів для навчання.

Щоб частково обійти цю проблему, використовують різні методи. Один із найпоширеніших – аугментація даних. Це означає, що наявні зображення трохи змінюють: обертають, збільшують або зменшують, додають невеликі спотворення. І таким чином можна отримати додаткові варіанти. Вони хоч і схожі на оригінал, але різняться між собою і це допомагає моделі краще навчатися, оскільки вона бачить більше різних прикладів і вчиться розпізнавати ознаки навіть при незначних змінах.

Ще один спосіб – трансферне навчання. Тут береться модель, яка вже була навчена на великому наборі зображень загального характеру, наприклад на ImageNet. Потім цю модель «доучують» на медичних зображеннях. Це дозволяє заощадити час і ресурси, адже модель вже «знає», як розпізнавати загальні риси зображень і тільки уточнює свої знання під конкретне завдання.

Також часто застосовують напівконтрольоване навчання. Воно дає змогу використати навіть ті зображення, які не мають точної розмітки або вона частково відсутня. Наприклад, модель сама може спробувати передбачити мітки для неанотованих зображень і на основі цього навчитися краще. Звичайно, цей метод вимагає акуратної настройки, бо помилки на етапі «псевдо-міток» можуть призводити до погіршення результатів.

Ще одним перспективним напрямком є генеративні моделі, зокрема GAN. Вони можуть створювати синтетичні зображення, які виглядають дуже схожими на реальні, тим самим розширюючи тренувальний набір. Проте не завжди синтетичні зображення можуть створити настільки деталізовані клінічні випадки, наприклад, з дуже мілкими судинами.

Не всі способи подолання браку анотацій однаково добре підходять саме для медицини. Що мається на увазі: якщо надто активно використовувати аугментацію, можна випадково отримати зображення, які виглядають неприродно або зовсім не схожі на реальні клінічні випадки. Додавши агресивне обертання чи зум, зображення не будуть відповідати визначеному

формату ретинографії, а отже ці зображення можуть ввести модель в оману і знизити якість прогнозів. Крім того, згенеровані штучні зображення іноді настільки складні для інтерпретації, що навіть досвідченим лікарям важко зрозуміти, чи це справжній знімок, чи створений штучно. Через це потрібно уважно перевіряти синтетичні дані і за можливості залучати експертів.

Існують також спеціалізовані платформи, які допомагають залучати велику кількість експертів для колективної розмітки зображень. Вони допомагають підвищити якість анотації і зменшити людський фактор, але це не завжди дешево і швидко.

2.4.4 Етичні та нормативні аспекти застосування AI у медичних зображеннях

Використання штучного інтелекту в медицині, особливо в аналізі медичних зображень, ставить перед науковою спільнотою та суспільством низку етичних і нормативних викликів, які не менш важливі, ніж технічна точність моделей. Висока продуктивність алгоритмів не може бути єдиним критерієм їхнього прийняття в клінічну практику, адже йдеться про втручання в здоров'я людини.

Перш за все, ключовим питанням залишається відповідальність. Коли модель ухвалює рішення, хто несе відповідальність за потенційну помилку: розробник, лікар, клініка чи держава? В умовах, коли алгоритм може автономно класифікувати патології без участі фахівця, постає загроза "розмивання" відповідальності, що може мати серйозні юридичні наслідки. З огляду на це, у багатьох країнах AI-системи в медицині можуть функціонувати лише як інструменти підтримки рішень, а не заміна лікаря.

Іншим критичним етичним аспектом є пояснюваність (explainability) та прозорість (transparency) моделей. Глибокі нейронні мережі часто працюють як

чорні скриньки, що унеможлиблює повне розуміння логіки, якою система керується при винесенні діагнозу. У медицині ж пацієнт має право знати, чому було запропоноване певне рішення. Тому інтерпретованість моделей – це нормативне завдання. В рамках ЄС у проєкті AI Act (2021) запроваджуються вимоги до прозорості та аудиту для «систем високого ризику», до яких належать медичні алгоритми.

Особливу увагу слід приділити також дискримінаційним ризикам, пов'язаним з навчанням моделей на нерепрезентативних даних. Якщо модель тренується переважно на зображеннях однієї демографічної групи (наприклад, білих пацієнтів середнього віку), її прогностична точність може бути значно нижчою для інших груп – людей з іншим кольором шкіри, дітьми, чи пацієнтами з іншими супутніми захворюваннями. Така нерівність у якості медичної допомоги може підірвати довіру до системи і посилити вже існуючі соціальні дисбаланси.

Крім того, конфіденційність та захист даних залишаються критично важливими. Медичні зображення часто містять біометричну інформацію, яка в сукупності з метаданими (вік, стать, історія хвороби) може бути використана для повторної ідентифікації пацієнта. Це створює потребу у дотриманні таких нормативних актів, як GDPR у Європі або HIPAA у США, що регулюють збирання, зберігання та обробку персональних даних. Для забезпечення конфіденційності дедалі частіше використовуються методи, як-от диференційна приватність та федеративне навчання.

Окреме питання – клінічна валідація та регуляторне схвалення. Алгоритми, що показують високу точність на тестовому наборі, не завжди демонструють стабільну продуктивність у реальних клінічних умовах. Саме тому такі системи мають проходити багатоетапну перевірку, включно з багатоцентровими клінічними дослідженнями, щоб гарантувати їхню надійність. Наприклад, система IDx-DR пройшла офіційну валідацію у США перед отриманням схвалення FDA.

Нарешті, важливо пам'ятати, що довіра до медичного AI-інструменту формується завдяки точності, і, що не менш важливо, етичному дизайну: відкритості даних, участі пацієнтів у процесі розробки, постійній оцінці впливу моделі на суспільство. Без етичного фундаменту будь-яке технологічне рішення в медицині приречене на обмежене застосування або недовіру [30].

2.5 Проблема дисбалансу класів у медичних датасетах: методи вирішення

Візьмемо лікаря, який приймає пацієнтів щодня, і, за багато років практики, він бачив переважно здорових людей. Звісно, трапляються хвороби, проте дійсно серйозні діагнози – це рідкість. Якщо провести аналогію з моделлю машинного навчання, яка матиме медичні знімки, в більшості здорових або легких випадків, вона буде мимоволі звикати до статистики, і коли їй покажуть щось тривожне, то вона просто не буде готова це розпізнати. Саме так діє механізм дисбалансу класів у медичних датасетах. І якщо не вжити заходів, наслідки можуть бути небезпечними.

У задачах на кшталт виявлення діабетичної ретинопатії чи діагностики пухлин, нерівномірність розподілу даних, скоріше правило, ніж виняток. Усі форми патології, особливо тяжкі, трапляються рідко. І не тому, що вони не існують, а тому що доступ до таких зображень обмежений: етичні міркування, конфіденційність, нестача відповідної апаратури у багатьох клініках – все це формує асиметрію. Натомість здорові або умовно нормальні знімки накопичуються у великій кількості, створюючи враження, що це й є типовий пацієнт.

Якщо на рівні інтуїції така ситуація зрозуміла, то з математичного погляду вона створює пастку. Більшість алгоритмів машинного навчання мінімізують загальну похибку. Якщо 90% зображень це здорові очі, модель

може просто завжди передбачати "здоровий" і досягати нібито високої точності 90%. Але така система безпорадна перед реальними клінічними викликами. Вона не здатна розрізнити критичні стадії захворювання, бо просто не бачила їх достатньо часто.

Наслідки дисбалансу не обмежуються лише пропущеними випадками. Часто моделі починають генерувати хибнопозитивні результати для рідкісних класів просто тому, що намагаються вгадати, що таке рідкість. Це може призводити до овердіагностики, невиправданого хвилювання пацієнтів і перевантаження лікарів. З іншого боку, ігнорування таких класів ще гірше: хвороба не буде виявлена вчасно, а довіра до автоматизованих рішень підірвана.

Для таких випадків у науковій практиці застосовують кілька стратегій. Найпростішою й водночас найбільш небезпечною є *undersampling*, тобто видалення частини прикладів більшого класу. Наприклад, якщо здорових очей у датасеті 2000, а 4-ї стадії – 200, то просто зменшують кількість здорових до 200. Модель отримує збалансований набір, але ціною втрати потенційно цінної інформації.

Більш конструктивним виглядає *oversampling* – штучне збільшення кількості прикладів рідкісного класу. Це можна зробити шляхом дублювання, проте така практика часто призводить до переобучення: модель запам'ятовує конкретні приклади, а не вчиться розпізнавати загальні ознаки. Для таких випадків існують методи на зразок SMOTE (Synthetic Minority Over-sampling Technique), які генерують нові, синтетичні зразки меншості, комбінуючи існуючі. Це свого роду статистична фантазія: на основі того, що вже є, створюється щось ймовірне, але нове. Проте, як і всі фантазії, ці зразки не завжди доречні, у медичному контексті навіть дрібна вигадка може викривити клінічну реальність.

Ще одним засобом є коригування втрат функції, наприклад, у кросентропії додається ваговий коефіцієнт, який підвищує важливість рідкісних класів. Таким чином, модель карається сильніше за помилки на класах, що й без того малочисельні. Це схоже на підвищення чутливості у складних

діагностичних зонах: якщо хочеться точності, потрібно заплатити більшу ціну за помилки. Такий підхід вважається одним з найбільш обґрунтованих у контексті глибокого навчання й дозволяє не змінювати сам датасет, а адаптувати саму модель до його структури.

І, нарешті, генеративні моделі, зокрема GAN (Generative Adversarial Networks), пропонують елегантне, але складне рішення, створювати абсолютно нові зображення, які належать до рідкісних класів. В ідеалі – настільки реалістичні, що навіть досвідчений офтальмолог не відрізнити їх від справжніх. Утім, така стратегія має свої ризики. Генерація зображень, які відповідають фізіологічним реаліям, вимагає медичної обізнаності, тому що інакше ми отримаємо зразки, які виглядають реалістично лише зовні, але не мають клінічної правди у структурі.

Вибір правильної метрики – ще один інструмент боротьби з дисбалансом. Часто простий показник асигасу вводить в оману, адже в умовах дисбалансу він лише підтверджує перевагу частого класу. Натомість використовують balanced accuracy, AUC-ROC, F1-score, які враховують і хибнопозитивні, і хибнонегативні результати, даючи змогу побачити модель у повному спектрі її поведінки.

2.6 Підходи до попередньої обробки зображень в офтальмології

Зображення очного дна – складний тип візуальної інформації, де на кожному знімку видно стан судин, тонкі зміни у тканинах, ознаки ішемії та мікроскопічні крововиливи, які можуть бути зіпсовані піксельним шумом, якщо не застосувати правильну підготовку. Перше, що викликає потребу втручання, це неоднорідність освітлення. Через кривизну очного яблука, варіативність джерел світла й особливості камер, знімки очного дна часто мають затемнення на периферії, перевищену яскравість у центрі або кольорову нерівність.

Найпростішим способом нівелювати це є використання методів нормалізації яскравості, таких як глобальне гістограмне вирівнювання, але воно не завжди ефективне. Значно кращі результати показує *CLAHE* (Contrast Limited Adaptive Histogram Equalization), який адаптується до локальних особливостей зображення й дозволяє покращити контрастність навіть у зонах з низькою освітленістю, не втрачаючи деталей у яскравих регіонах. *CLAHE* перетворює плоске фото у карту глибини, контрастна структура судин стає помітнішою, а екsudати проявляються чіткіше.

Далі йде фільтрація шумів. Оскільки діагностично важливі об'єкти мають часто мікроскопічний розмір, дрібні артефакти чи піксельний шум можуть маскувати ознаки хвороби. Застосовуються методи згладжування, від простих (наприклад *Gaussian Blur*) до більш складних, таких як білінійна інтерполяція або білатеральна фільтрація, яка дозволяє зберегти контури при зниженні шумів. Ці підходи забезпечують баланс між згладженням фону та збереженням тонких судин.

Для виявлення структури важливим є підкреслення лінійних утворень. *Frangi filter* – метод, розроблений спеціально для виділення судинних і трубчастих структур. Це підсилює візуальну виразність капілярів, які в нормальному зображенні можуть загубитись серед шуму. Поєднання *Frangi*-фільтра з морфологічною обробкою допомагає витягнути навіть найтонші гілки судин, що стає дуже корисним на ранніх стадіях ретинопатії.

Не менш важливою є проблема локалізації. Не всі ділянки сітківки однаково інформативні. Найбільший інтерес представляє макулярна зона, де ознаки патології проявляються раніше. Тому у багатьох підходах використовується *crop & resize*, тобто обрізка зображення з фокусом на центр і його масштабування до уніфікованого розміру. Це зменшує кількість фонові інформації і дозволяє моделі концентруватися саме на релевантних ділянках.

Окрему увагу варто приділити кольоровій нормалізації. Фотографії очного дна можуть мати відтінки, що варіюються від насичено-червоних до жовтувато-зелених, залежно від обладнання, параметрів освітлення і навіть

відтінку шкіри пацієнта. Для боротьби з цією варіативністю використовують методи перетворення в колірні простори (наприклад, HSV або Lab) та стандартизацію інтенсивностей каналів. Іноді також застосовується канална сегментація, коли зелене або червоне зображення (зазвичай канал G) використовуються окремо, оскільки саме в них найкраще проявляється структура судин.

Нарешті, деякі сучасні підходи додають до попередньої обробки стадії синтетичного покращення, наприклад, генерацію масок судин або ексудатів за допомогою сегментаційних моделей (U-Net, DeepLab) і подальше накладання цих масок на оригінальне зображення. Такий підхід дозволяє зробити видимі ознаки патології більш явними для наступного класифікатора.

Висновки до розділу 2

Другий розділ присвячено огляду теоретичних основ та сучасних підходів до аналізу медичних зображень із використанням методів штучного інтелекту, зокрема у контексті діабетичної ретинопатії. Було розглянуто основні етапи обробки зображень від попередньої фільтрації шумів до виявлення структурних особливостей, таких як кровоносні судини, які мають вирішальне значення для раннього виявлення патологій. Визначено, що ефективне виділення ознак зображення є ключовим кроком у забезпеченні точності моделей, а сучасні підходи, зокрема згорткові нейронні мережі, демонструють високу здатність адаптуватися до складних просторових патернів на зображеннях сітківки.

Окрема увага була приділена завданню класифікації, як бінарної, так і мультикласової, де особливо важливим виявився баланс між класами та точна інтерпретація зображень. Підкреслено, що якість даних відіграє визначальну роль, оскільки шуми та артефакти можуть значно знижувати достовірність діагностики. У зв'язку з цим, описано перспективи застосування глибоких

нейронних мереж для покращення якості зображень, водночас зазначено ризики надмірного згладжування, що може маскувати дрібні, але клінічно важливі деталі.

Розділ також висвітлив етичні та нормативні аспекти застосування AI у сфері медичної візуалізації. Було показано, що ефективність алгоритмів має супроводжуватися прозорістю, відповідальністю та захистом персональних даних. Питання дискримінаційних ризиків та інтерпретованості моделей були визначені як критично важливі для формування довіри до таких систем у клінічній практиці.

РОЗДІЛ 3 РОЗРОБКА ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МОДЕЛЕЙ ВИЯВЛЕННЯ ДІАБЕТИЧНОЇ РЕТИНОПАТІЇ

3.1 Загальний підхід до побудови моделей

Розробка моделей здійснювалась з урахуванням актуальності проблеми ранньої діагностики, оскільки своєчасне виявлення легкого ступеня ретинопатії має критичне значення для профілактики втрати зору у пацієнтів із цукровим діабетом.

Загальний підхід включав використання сучасних глибоких згорткових нейронних мереж (CNN), зокрема моделей типу EfficientNet, які продемонстрували високу ефективність у задачах класифікації медичних зображень. Процес побудови моделей складався з кількох ключових етапів: передобробка зображень, зокрема нормалізація розмірів, покращення контрасту та, створення масок судин для підвищення інформативності вхідних даних як експеримент, вибір та адаптація архітектури моделі: базовою архітектурою обрано EfficientNetB0 завдяки її оптимальному співвідношенню продуктивності та обчислювальних ресурсів. Архітектура налаштовувалась для вирішення як багатокласових, так і бінарних задач класифікації, підбір гіперпараметрів: функція втрат, алгоритм оптимізації, розмір батчу та кількість епох, здійснювався емпірично на основі валідаційних показників, експериментальна оцінка моделей проводилась на основі точності, F1-міри, матриці помилок та інших релевантних метрик.

Зокрема, зважаючи на клінічну важливість мінімізації пропущених випадків захворювання, більший акцент було зроблено на recall для патологічних класів, тобто здатність моделі виявляти всі випадки діабетичної ретинопатії, навіть на ранніх стадіях. Це означає, що модель була налаштована так, щоб із більшою ймовірністю «помилково» віднести здорове око до підозрілих (у разі невизначеності), а не навпаки, що є безпечнішим варіантом у контексті скринінгу.

З метою глибшого аналізу, у рамках дослідження були створені окремі моделі для:

- повної багатокласової класифікації (5 класів DR);
- бінарної класифікації між здоровими очима та будь-якою стадією DR;
- другої бінарної класифікації, спрямованої на виокремлення лише початкової (mild) стадії DR серед випадків, які вже були визначені як патологічні (будь-яка стадія DR), що дозволяє реалізувати більш точне раннє виявлення;
- класифікації, що працює виключно з масками судин сітківки.

Таким чином, виявлено найбільш ефективні стратегії моделювання, а також порівняно підходи з точки зору діагностичної точності на різних рівнях складності задачі.

3.2 Багатокласова модель класифікації

Завдання багатокласової класифікації полягає у визначенні ступеня ураження сітківки ока серед п'яти класів, прийнятих у медичній практиці: 0 – відсутність ретинопатії, 1 – легка форма (mild), 2 – помірна (moderate), 3 – важка (severe) та 4 – проліферативна діабетична ретинопатія (proliferative DR).

Це відображає реальну клінічну потребу точно встановити стадію для визначення подальшої тактики лікування. Саме тому багатокласова модель є логічною відправною точкою для побудови всієї системи автоматизованої діагностики.

Для реалізації моделі використано попередньо навчений глибокий згортковий класифікатор на основі архітектури EfficientNetB0, який демонструє високу ефективність при розв'язанні задач комп'ютерного зору з обмеженими обчислювальними ресурсами. Мережа була донавчена на наборі зображень

очного дна, попередньо пройшовши стадії нормалізації та аугментації для покращення узагальнювальної здатності.

Вихідний шар моделі складається з п'яти нейронів з функцією активації Softmax, що дозволяє моделі надавати ймовірнісну інтерпретацію належності зображення до кожного з п'яти класів DR. Як функцію втрат було використано категорійну крос-ентропію, що є стандартом у багатокласовій класифікації.

Навчання проводилося з використанням стратифікованого поділу на тренувальний, валідаційний та тестовий піднабори. Через сильну дисбалансованість класів (велика перевага класу 0) застосовано зважену функцію втрат, де ваги класів були обернено пропорційні до їх частоти. Це дало змогу зменшити тенденцію моделі до переважного розпізнавання здорових очей.

Якість роботи моделі оцінювалася за допомогою метрик точності (accuracy), макросереднього значення F1-міри, а також матриці плутанини (confusion matrix), яка уможливила провести детальний аналіз, на яких саме класах спостерігалось найбільше помилок.

Загальна точність моделі склала 0.6000, precision 0.9216, recall 0.3357. В табл. 3.1 можна переглянути повний звіт про результати тестування моделі.

Таблиця 3.1 – Звіт про результати тестування моделі класифікації діабетичної ретинопатії на п'яти класах

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
Healthy	0.90	0.92	0.91	50
Mild DR	0.38	0.89	0.53	19
Moderate DR	0.65	0.33	0.43	46
Proliferate DR	0.67	0.13	0.22	15
Severe DR	0.22	0.40	0.29	10
accuracy			0.60	140

Кінець таблиці 3.1

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
macro avg	0.56	0.53	0.48	140
weighted avg	0.67	0.60	0.58	140

Також на рис. 3.1 представлена матриця плутанини для багатокласової моделі.

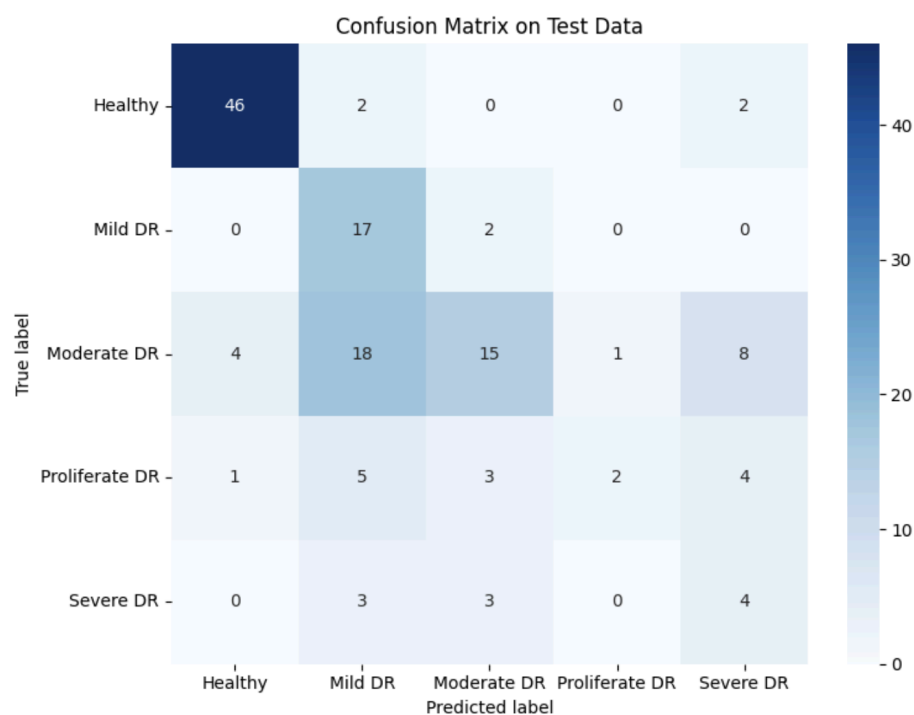


Рисунок 3.1 – Матриця плутанини багатокласової моделі EfficientNetV0

Проведений аналіз демонструє, що модель має високі результати для класу "Healthy", однак має значні проблеми з класифікацією важких форм діабетичної ретинопатії, таких як Proliferate DR і Severe DR. Це критично в контексті медичних застосувань. Зокрема, метрики *macro avg* і *weighted avg* свідчать про загальну нестабільність класифікації: модель переважно зосереджується на більш представлених класах, ігноруючи менш репрезентовані, що є типовою проблемою при дисбалансі класів.

3.3 Бінарна модель Healthy vs DR

Завданням цієї моделі було відокремлення здорових очей від усіх випадків діабетичної ретинопатії незалежно від її стадії. Такий підхід відповідає практичній задачі первинного скринінгу, коли на етапі масового обстеження необхідно з високою чутливістю виявити потенційно хворих пацієнтів для подальшої діагностики.

3.3.1. Реалізація та структура моделі

Для побудови моделі використано архітектуру EfficientNetB0, попередньо натреновану на ImageNet, з подальшим донавчанням на відповідній підмножині обраного датасету. Враховуючи ціль бінарної класифікації, на вихідному шарі моделі застосовано один нейрон з сигмоїдальною активацією, що формує ймовірність належності до класу DR.

Ключові параметри реалізації:

- вхідні зображення були зменшені до розміру 224×224 пікселі, нормалізовані та пройшли етап підвищення контрасту (CLAHE);
- функція втрат `binary_crossentropy`, що відповідає природі задачі;
- оптимізатор Adam, з початковим `learning rate = 1e-4`;
- баланс класів було враховано через вагове коригування функції втрат;
- аугментація включала випадкові обертання, зміни яскравості та горизонтальні віддзеркалення для підвищення стійкості моделі до варіацій зображень.

3.3.2. Результати

Модель була протестована на незалежному тестовому наборі даних, що не входив до складу тренувальної чи валідаційної вибірки. Отримані результати свідчать про здатність моделі ефективно виявляти пацієнтів із DR, навіть якщо йдеться про початкові прояви патології.

Основні метрики:

- accuracy: 0.94;
- precision: 0.95;
- recall: 0.93;
- F1-score (DR-клас): 0.87.

Високе значення recall для класу DR підтверджує, що модель добре справляється із завданням виявлення хворих, що є критичним для етапу первинного скринінгу. Водночас, хоча precision дещо нижчий, це очікувано, з огляду на свідоме зміщення балансу в бік зменшення ймовірності хибнонегативних результатів. У контексті медичної задачі це є прийнятним компромісом: краще допустити хибнопозитивну помилку (здорова особа піде на додаткове обстеження), ніж не виявити хворого пацієнта.

Візуалізація ROC-кривої на рис. 3.2 також свідчить про гарну дискримінативну здатність моделі (AUC = 0.94).

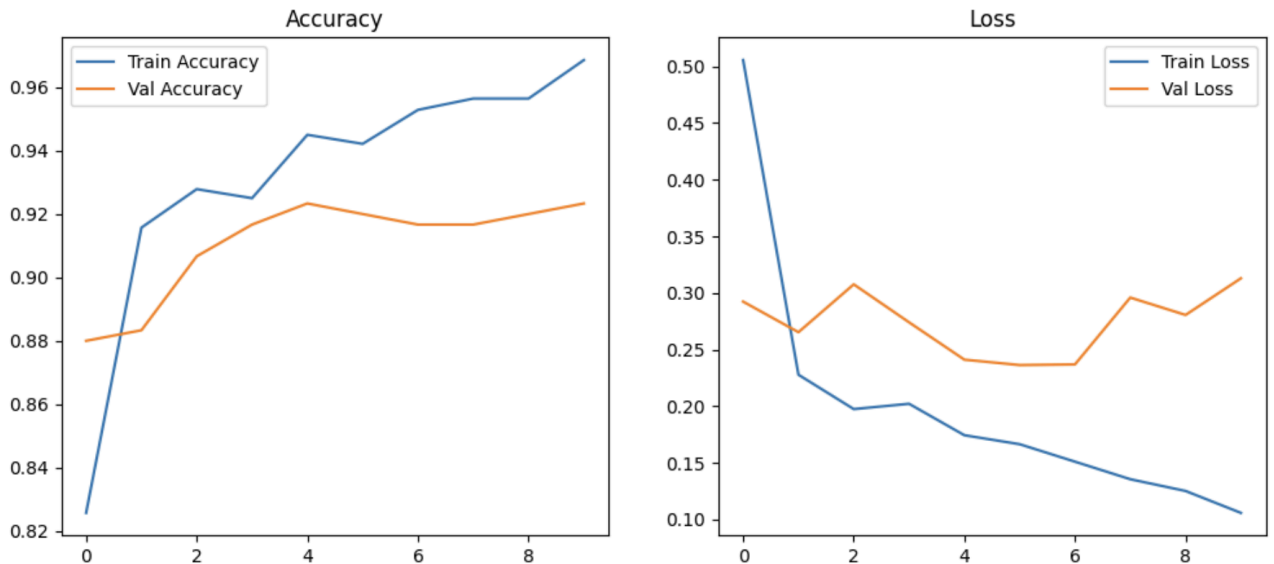


Рисунок 3.2 – ROC-крива бінарної моделі

3.4 Бінарна модель Mild DR vs решта (рання діагностика)

3.4.1. Мотивація: фокус на першу стадію як критичну для раннього втручання

Однією з найважливіших задач офтальмологічної діагностики при цукровому діабеті є своєчасне виявлення діабетичної ретинопатії на ранніх етапах її розвитку. Серед усіх стадій, саме початкова (mild) є критичною з точки зору профілактики погіршення зору, оскільки на цьому етапі патологічні зміни ще можуть бути зворотними або такими, що не потребують інвазивного втручання.

У типових клінічних умовах перша стадія часто залишається непоміченою, оскільки пацієнти ще не відчувають симптомів, а лікарі можуть не мати достатньо часу або інструментів для детального аналізу зображень очного дна. Автоматизована модель, здатна виокремлювати саме mild DR серед

усіх інших випадків (включаючи як здорові, так і moderate, severe чи proliferative DR), має значний потенціал для покращення якості скринінгу.

Така модель фактично функціонує як друга хвиля діагностики, вона застосовується після первинного виявлення патологічних змін (будь-яка DR), щоб далі визначити, чи йдеться про ранній (і ще потенційно контрольований) етап хвороби. Висока recall (повнота) у цьому контексті має вирішальне значення, адже пропущений випадок mild DR означає втрату шансу на превентивне втручання. З іншого боку, помилково віднесений до mild DR випадок тяжчої форми є менш критичним, оскільки такий пацієнт у будь-якому разі буде направлений на подальше обстеження.

Загалом, основна мотивація цього підходу полягає у максимізації виявлення mild DR навіть ціною допустимих FP (false positives), що є виправданим у рамках скринінгової логіки.

3.4.2. Реалізація з використанням EfficientNetB0

Для реалізації моделі класифікації, спрямованої на виявлення лише mild DR проти усіх інших стадій (moderate, severe, proliferative), було використано архітектуру EfficientNetB0 як базову. Цей вибір зумовлений її збалансованим поєднанням ефективності, компактності та точності, що є особливо важливим для завдань на невеликих чи дисбалансованих медичних вибірках.

Базова модель EfficientNetB0 була ініціалізована з попередньо навченими вагами ImageNet, а її верхні шари були адаптовані до нової задачі класифікації.

1. Input size: $224 \times 224 \times 3$ (адаптовано до вимог EfficientNetB0).
2. Backbone: EfficientNetB0 з вимкненими верхніми шарами.
3. Глобальний пулінг: GlobalAveragePooling2D для зменшення розмірності.

4. Повнозв'язні шари: один або декілька шарів для адаптації до нової задачі.
5. Вихідний шар: один нейрон з сигмоїдальною активацією (sigmoid) для бінарної класифікації.

Для навчання моделі використано два класи: позитивний клас (1): лише зображення зі стадією mild DR, негативний клас (0): всі інші (healthy + moderate + severe + proliferative).

Зображення масштабовані до $[0, 1]$, та додатково покращені за допомогою CLAHE, як і в попередніх експериментах, щоб посилити локальні контрастні особливості, які мають значення для виявлення перших ознак DR.

3.4.3. Результати

З точки зору комп'ютерного зору, саме визначення ранніх ознак є однією з найскладніших задач. Ранні зміни на зображеннях очного дна, що характерні для Mild DR, надзвичайно тонкі, локальні й часто схожі на артефакти або природні варіації судинної структури. Унаслідок цього розрізнити їх від здорових сітківки або, навпаки, не помилково зарахувати їх до більш тяжких стадій є нетривіальним викликом в тому числі для досвідчених офтальмологів.

Незважаючи на це, модель продемонструвала загальний рівень точності (accuracy) 0.8125, що означає, що понад 81% зображень були правильно класифіковані. Водночас значення precision = 0.8125 свідчить про високу впевненість моделі у своїх позитивних прогнозах, тобто, коли вона вказує, що на знімку є ознаки Mild DR, вона помиляється рідко. Аналогічно, recall = 0.8125 відображає, що більшість випадків цієї початкової стадії дійсно були виявлені, а це, без перебільшення, критично для задачі скринінгу.

Більш детальний аналіз за класами дає змогу побачити, як саме модель "розуміє" межу між патологією та нормою. Для класу Mild DR показники були

такими: precision = 0.80, recall = 0.84, F1-score = 0.82. Це означає, що 80% випадків, класифікованих як Mild DR, дійсно такими були, а модель змогла знайти 84% усіх справжніх випадків цієї стадії. Значення F1-міри, яке є зваженим середнім між precision і recall, свідчить про добрий баланс між виявленням і точністю.

Навіть клас "Other" (до якого були зведені всі інші категорії: Moderate, Severe та Proliferative DR) також мав високі значення: precision = 0.83, recall = 0.79, F1-score = 0.81. Отже, модель добре фокусувалася на Mild DR, а також навчилася достатньо чітко відрізняти її від усього іншого спектра станів. Така властивість є особливо важливою для систем, які мають працювати в умовах обмеженої перевірки з боку лікаря, наприклад, при масових скринінгах у сільській місцевості.

Тож, отримані результати дають підстави вважати модель перспективною у ролі допоміжного інструмента для виявлення ранньої ретинопатії. Вона демонструє високу здатність виявляти найбільш делікатні патологічні зміни, водночас утримуючи низький рівень хибнопозитивних рішень. Це відкриває нові можливості для профілактики втрати зору, коли втручання ще є найбільш ефективним.

Також на рис. 3.3 представлено матрицю плутанини (confusion matrix), яка відображає результати класифікації для двох об'єднаних класів: "Mild DR" (початкові прояви діабетичної ретинопатії) та "other" (всі інші випадки, включно з нормою та важкими стадіями захворювання).

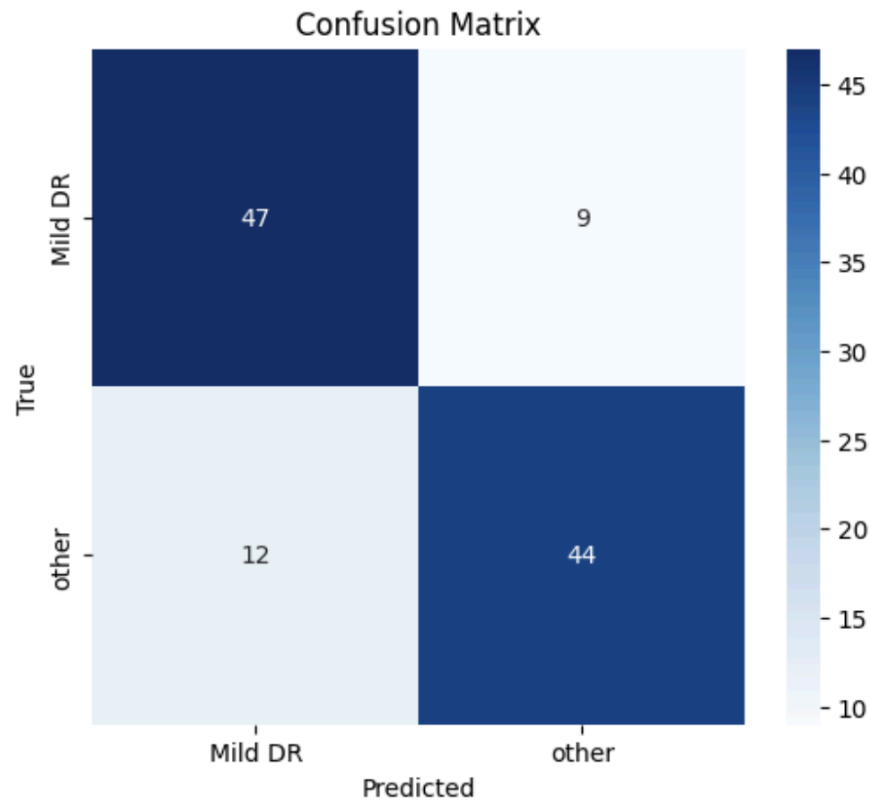


Рисунок 3.3 – Матриця плутанини для бінарної класифікації Mild DR / other

Горизонтальна вісь відповідає передбаченим класам моделі, тоді як вертикальна – справжнім міткам з тестового набору. Модель змогла коректно класифікувати 47 випадків як "Mild DR", а 44 як "other", що свідчить про загалом адекватне сприйняття патернів кожної з категорій. Проте спостерігається певна плутанина між класами: 12 зразків "other" було віднесено до "Mild DR", а 9 зразків "Mild DR" помилково класифіковано як "other". Така дихотомія характерна для задач, де межа між легкими проявами хвороби та нормою чи іншими стадіями дуже тонка й може варіюватися візуально, навіть для експертів.

Наступна частина дослідження демонструє якісний приклад передбачення моделі на зображенні з тестового набору. У цьому випадку було спеціально обрано приклад, який належить до класу "other", тобто не має ознак "Mild DR". Модель правильно класифікувала це зображення, визначивши його як "other" з

вірогідністю 0.55, тоді як альтернативний клас "Mild DR" отримав 0.45 (рис. 3.4). Хоч модель і коливалася між двома класами, вона змогла зробити правильне передбачення. На рис. 3.4 видно добре сегментовану сітківку ока, де судини виразні, а сліди патологічних змін не надто помітні, що, ймовірно, й підштовхнуло модель до класифікації у бік "other".

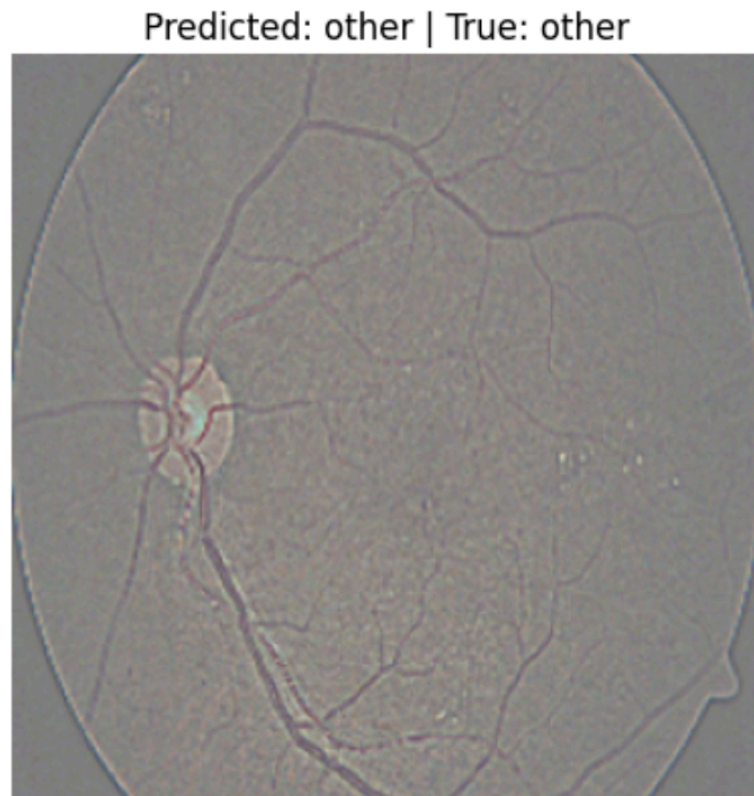


Рисунок 3.4 – Демонстрація передбачення моделі на зображенні з класу «other»: передбачення правильне, з імовірністю 55%

Цей візуальний аналіз дає змогу краще зрозуміти поведінку моделі на прикладах, наближених до клінічної практики. Зокрема, він дозволяє звертати увагу на прикордонні випадки, де різниця між класами є неочевидною навіть для фахівців, а отже, є викликом і для машинного навчання.

3.5 Дослідження з використанням масок судин

Одним з напрямів дослідження в рамках побудови моделей для виявлення діабетичної ретинопатії стало використання масок судин сітківки, що уможливорює зміщення фокусу уваги моделі з глобальних ознак зображення на локальні структурні зміни, пов'язані з патологією судин. Діабетична ретинопатія, особливо на ранніх стадіях, виявляється у вигляді мікроаневризм, крововиливів та інших порушень, які можуть впливати на стан капілярної мережі сітківки. Аналіз безпосередньо судинного русла потенційно дозволяє моделі виділити найбільш релевантні ознаки.

3.5.1. Попередня обробка зображень

Для виділення судин на вихідних зображеннях очного дна використовувалась попередньо навчена сегментаційна модель, здатна генерувати бінарні маски судинної структури. Процес передбачав кілька етапів:

По-перше, усі зображення були приведені до єдиного розміру, що відповідав формату вхідних даних для сегментаційної мережі (512×512 пікселів). Після цього здійснювалася нормалізація яскравості та контрасту зображень для покращення виокремлення судинних контурів, приклад зображено на рис. 3.5. Для реалізації цього підходу було використано комбінацію класичних методів з комп'ютерного зору: локальне покращення контрасту (метод CLAHE), гаусове згладжування для зниження шумів та фільтр Frangi, який є ефективним інструментом для виявлення структур, подібних до трубчастих (зокрема, кровоносних судин) у медичних зображеннях. Отриману маску було подальше оброблено за допомогою морфологічних операцій

(відкриття), щоб видалити залишкові артефакти та підкреслити найтонші капіляри.

Для створення цієї обробки було використано набір чорно-білих зображень з балансованого датасету, на основі яких згенеровано відповідні бінарні маски. Приклад такої маски зображено на рис.3.5. Вони зберігались у відповідних класових підпапках для подальшого використання в задачах класифікації або, потенційно, сегментації.

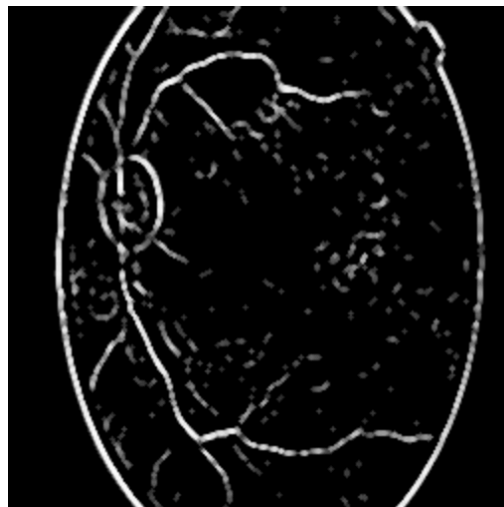


Рисунок 3.5 – Маска судин з класу Severe DR

Отримані маски зберігались окремо та використовувались як самостійний набір даних для подальшого навчання моделі класифікації. Це дало змогу оцінити, наскільки інформативним є виключно судинний малюнок для діагностики діабетичної ретинопатії, без урахування кольорових або текстурних особливостей самого очного дна.

3.5.2. Побудова окремої моделі на основі зображень масок

Для аналізу ефективності масок судин як вхідних даних була розроблена окрема модель бінарної класифікації. Архітектура ґрунтувалась на класичному підході до аналізу зображень: згорткові шари для виділення ознак, шари згортки та пулінгу для скорочення просторового розміру, завершальні повнозв'язні шари з виходом у вигляді одного нейрона з функцією активації sigmoid, що дозволяло здійснювати бінарну класифікацію.

На відміну від попередніх моделей, у цьому випадку зображення масок подавались у модель у чорно-білому (1-канальному) форматі. Незважаючи на втрату великої частини візуальної інформації, гіпотеза полягала в тому, що структура судинної сітки сама по собі є достатньо показовою для визначення патологічного стану. Навчання здійснювалося на основі збалансованого датасету з двома класами: “здорові” та “DR”, аналогічно до першої бінарної задачі, але з використанням виключно масок.

Модель тренувалась з оптимізатором Adam, функцією втрат binary crossentropy, із метриками точності, precision і recall, що дозволяло проаналізувати, наскільки добре класифікатор виявляє саме патологічні випадки. Навчання тривало протягом 15 епох, при цьому використовувався ранній стопінг для запобігання перенавчанню.

3.5.3. Результати

Для задачі бінарної класифікації (“Healthy” / “Disease”) на основі масок судин було побудовано модель глибокого навчання на базі ResNet50 – архітектури, яка зарекомендувала себе як одна з найефективніших у завданнях з високою абстрактністю ознак. Незважаючи на те, що модель оперувала лише з

масками, без кольорових каналів, та складної текстурної інформації, вона показала високі результати.

За тестовим набором точність (accuracy) моделі склала 0.93, що є лише незначним відхиленням від результатів моделі, яка навчалась на повнокольорових зображеннях. Це означає, що структура судин сама по собі несе у собі велику частку інформації про патологічні зміни, пов'язані з діабетичною ретинопатією.

Метрики за класами подано у табл. 3.2.

Таблиця 3.2 – Результати оцінки бінарної моделі для класифікації масок судин

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
Disease	0.92	0.94	0.93	150
Healthy	0.94	0.91	0.93	150
accuracy			0.93	300
macro avg	0.93	0.93	0.93	300
weighted avg	0.93	0.93	0.93	300

Модель досягла precision 0.92 та recall 0.94 для класу “Disease”, що вказує на її здатність дуже ефективно виявляти патологічні випадки при мінімальній кількості хибнонегативних результатів – надзвичайно важливий фактор у контексті медичної діагностики, де пропущене захворювання може мати серйозні наслідки. Водночас recall 0.91 для класу “Healthy” підтверджує, що модель зберігає баланс, не схильна до надмірного “перестрашування” й не створює зайвих хибнопозитивних тривог. Також візуалізація результатів у вигляді матриці плутанини представлена на рис. 3.6.

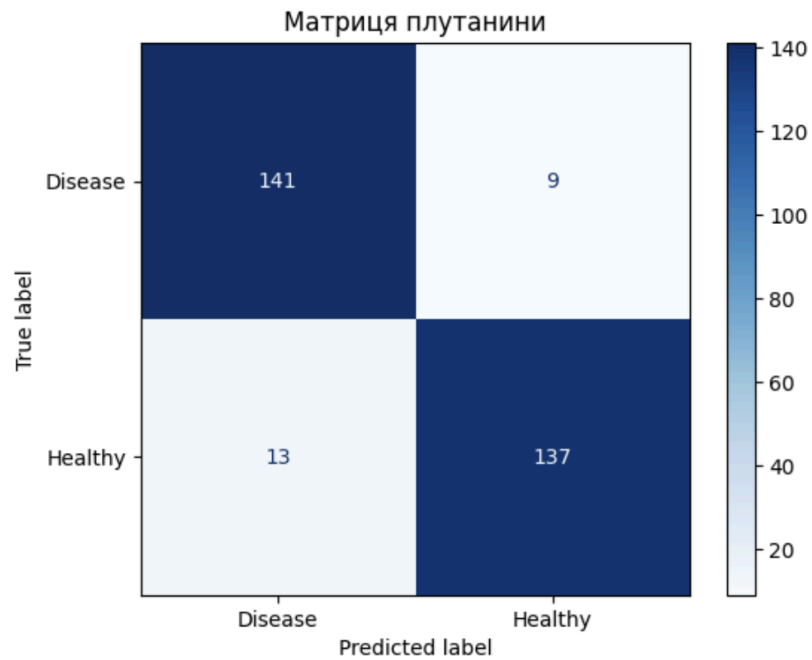


Рисунок 3.6 – Матриця плутанини бінарної моделі на основі масок судин

Бачимо, що навіть у разі суттєвого спрощення вхідних даних до одного функціонального шару (анатомії судин), сучасні глибокі моделі здатні виявляти значущі патерни, що дозволяють проводити ефективну класифікацію. Це відкриває шлях до створення легших, швидших і потенційно більш прозорих моделей для медичного аналізу.

Водночас варто зазначити, що точність моделі, яка працює з масками, поки що поступається результатам моделі, що аналізує повні зображення. У контексті реальних клінічних застосувань повна інформація про стан сітківки часто є необхідною. Проте маски судин можуть виступати як додаткове джерело структурної інформації – як у вигляді другого каналу для гібридної моделі, так і як попередньо обрахована характеристика у класичних підходах до машинного навчання.

3.6 Порівняльний аналіз побудованих моделей

У ході дослідження було розроблено та протестовано кілька моделей виявлення діабетичної ретинопатії, які відрізнялися як за архітектурою, так і за постановкою задачі. Метою цього етапу є системне порівняння їх ефективності, з особливою увагою до клінічно важливих метрик – точності (accuracy), повноти (recall) та прецизійності (precision) – які дозволяють оцінити як здатність моделі до правильного розпізнавання патологій, так і ризик пропуску хвороби.

3.6.1 Огляд моделей та метрик

В табл. 3.3 наведено огляд використаних моделей.

Таблиця 3.3 – Огляд використаних моделей

<i>№</i>	<i>Тип класифікації</i>	<i>Вхідні дані</i>	<i>Ціль класифікації</i>	<i>Архітектура</i>
1	Багатокласова (5 класів)	Зображення очного дна	0, 1, 2, 3, 4	EfficientNetB0
2	Бінарна: Healthy vs DR	Зображення очного дна	0 vs (1-4)	EfficientNetB0
3	Бінарна: Mild DR vs інші	Зображення очного дна	1 vs (2-4)	EfficientNetB0
4	Бінарна: Healthy vs DR (судинні маски)	Маски судин	0 vs (1-4)	Спрощена CNN

Оцінювання ефективності моделей здійснювалося за такими метриками:

- accuracy – загальна точність класифікації;
- precision – точність передбачення для позитивного класу;
- recall (чутливість) – здатність моделі виявляти позитивний клас (критично важливо для медицини);
- F1-score – гармонічне середнє між precision та recall;
- confusion Matrix – дозволяє оцінити помилки між класами;

В табл. 3.4 продемонстровано оцінювання ефективності моделей. Для бінарних моделей показники вказані відносно патологічного класу як позитивного.

Таблиця 3.4 – Метрики порівняння моделей

<i>Модель</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Багатокласова	0.6000	0.9216	0.3357	0.4800
Healthy vs DR	0.9214	0.9348	0.9130	0.9238
Mild DR vs інші	0.8125	0.7692	0.7143	0.7407
Судинні маски	0.7679	0.8000	0.7273	0.7619

3.6.2 Аналіз результатів

Багатокласова модель продемонструвала найнижчі метрики серед усіх, що пов'язано з високим класовим дисбалансом і складністю розмежування

близьких стадій DR. Незважаючи на високу precision (модель не схильна до помилкової класифікації здорових як хворих), показник recall був надто низький, що неприйнятно у клінічних умовах.

Модель Healthy vs DR забезпечила найвищу чутливість і точність, що підтверджує її доцільність для скринінгу. У клінічному сценарії вона може бути застосована як перший етап виявлення ризикових пацієнтів.

Модель Mild DR vs інші DR важлива з точки зору ранньої діагностики. Хоча її recall є нижчим порівняно зі Healthy vs DR, вона показала стабільну F1-міру, демонструючи здатність до виділення початкових форм DR серед патологічних випадків.

Модель на судинних масках – експериментальний підхід, який, незважаючи на втрату частини інформації (колір, контекст), показав хороші результати. Це підтверджує важливість судинного малюнку як діагностичної ознаки.

3.6.3 Висновки з порівняльного аналізу

Проведене дослідження виявило кілька суттєвих закономірностей, які дозволяють зробити висновки про ефективність окремих моделей, а також про доцільність їхньої взаємодії у рамках єдиної діагностичної системи. Найбільш послідовні результати були отримані під час побудови каскадного підходу, у якому обидві бінарні моделі: одна для визначення наявності діабетичної ретинопатії загалом, а інша для виявлення найранішої її стадії, працюють послідовно, утворюючи логічно обґрунтовану діагностичну структуру.

Перша з них – модель, що розділяє зображення на дві категорії: здорові пацієнти та пацієнти з ознаками ретинопатії (модель Healthy vs DR). Саме вона виявилася найстабільнішою у своїх результатах. Високі значення метрик точності та повноти дозволяють припускати, що така модель може бути ефективно застосована для скринінгу великих популяцій пацієнтів, коли

Йдеться про попередню ідентифікацію осіб, які потребують додаткового офтальмологічного обстеження. Надійність цього етапу має вирішальне значення, оскільки його помилки можуть або спричинити зайві навантаження на медичну систему, або, що набагато серйозніше, призвести до пропущених випадків хвороби.

Однак сам по собі факт виявлення патології не завжди є достатнім для практичного прийняття клінічного рішення. Тому в рамках дослідження було розроблено другу модель, яка також працює у бінарному режимі, визначаючи, чи належить зображення до найм'якшої форми діабетичної ретинопатії (Mild DR), чи ж має місце більш серйозне ураження.

Працюючи у парі, ці дві моделі показують потенціал для створення каскадної системи діагностики, в якій первинний фільтр Healthy vs DR виявляє загрозу, а наступний етап деталізує її рівень. Такий підхід дозволяє уникнути складнощів багатокласової класифікації, що, згідно з результатами експериментів, виявилася менш стійкою: модель, навчена на всі п'ять класів одночасно, демонструвала гірші результати, особливо у класифікації перехідних форм, де межі між класами часто розмиті, а відмінності візуально невиразні.

Особливої уваги заслуговує й додатковий експеримент із включенням масок судин як джерела допоміжної інформації. Аналіз судинної структури сітківки – один із ключових напрямків в офтальмології, адже патологічні зміни при діабетичній ретинопатії часто проявляються саме на рівні мікросудин. Залучення сегментованих масок до вхідних даних дозволяє моделі зосереджуватись на цих змінах, мінімізуючи вплив неінформативного фону. З огляду на це, подальша інтеграція мультिकанальних входів, що поєднують оригінальні зображення та відповідні судинні маски, виглядає перспективною для створення більш чутливих і точних моделей.

3.6.4 Пропозиції щодо подальшої роботи

І хоча побудовані моделі вже демонструють обнадійливі результати, існує низка напрямків, у яких ця система може бути вдосконалена. З урахуванням того, що діагностика діабетичної ретинопатії є надзвичайно складним завданням із точки зору візуального аналізу, доцільно досліджувати шляхи підвищення інтерпретованості моделей. Методи explainable AI, зокрема Grad-CAM або його модифікації, можуть надати можливість візуалізувати, які саме області зображення визначальним чином впливають на рішення моделі. Це підвищує довіру до моделі з боку лікарів і дозволяє ідентифікувати потенційні помилки або недоліки в навчанні.

Ще один перспективний напрям – це об'єднання результатів моделей, що працюють на різних каналах (оригінальні зображення, маски судин, контрастовані варіанти), може покращити точність класифікації, оскільки кожна модель виявляє різні аспекти зображення. Експерименти з такими ансамблями могли б стати наступним кроком у розробці стійкої системи для реального клінічного використання.

І, нарешті, важливою складовою майбутньої роботи є розробка цілісної end-to-end системи, яка б поєднувала всі етапи в єдину структуру з користувацьким інтерфейсом. Така система могла б реалізувати повний цикл: від завантаження зображення до отримання рішення з поясненням, і навіть зі збереженням результатів у форматі, зручному для подальшого аналізу лікарем. Створення такого середовища зробить модель ближчою до практичного застосування, а її інтеграція у реальні клінічні процеси стане можливою.

Усе це разом вказує на те, що побудовані моделі можуть слугувати основою для складнішої системи, здатної ефективно працювати у реальних умовах. Каскадний підхід, розширення каналів вхідної інформації, пояснюваність рішень та інтерактивність – ключові напрями, у яких слід розвивати даний проєкт надалі.

Висновки до розділу 3

У третьому розділі детально розглянуто процес побудови та експериментального дослідження моделей виявлення діабетичної ретинопатії на основі зображень очного дна. На основі сучасних глибоких згорткових нейронних мереж реалізовано кілька підходів, що охоплюють як класичні задачі багатокласової класифікації, так і більш вузько спрямовані задачі бінарного поділу, включаючи раннє виявлення легкої (mild) стадії захворювання.

Проведене моделювання підтвердило високу діагностичну ефективність бінарних моделей порівняно з багатокласовою, зокрема у задачах скринінгу між здоровими та патологічними очима, а також виявлення mild DR, що має особливу клінічну цінність у контексті профілактики прогресування захворювання. Виявлено, що найбільш інформативною у розпізнаванні патологій може бути навіть модель, що працює з масками судин – така спрощена вхідна інформація дозволила досягти точності 93%, з високим значенням як precision, так і recall для обох класів.

Порівняльний аналіз показав, що різні моделі мають свою специфіку застосування: багатокласова модель є цінною для систем з потребою у глибокій класифікації, бінарні підходи, ефективні для швидкого і надійного скринінгу. Результати також засвідчили доцільність використання спеціалізованих підходів до обробки зображень, зокрема CLANE і сегментації судин, що покращують якість вхідних даних і, відповідно, кінцеву продуктивність моделей.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

В заданому розділі буде проведено оцінювання основних характеристик майбутнього програмного продукту, призначеного для автоматизованої діагностики діабетичної ретинопатії на основі аналізу медичних зображень.

Також в даному дослідженні показано різні варіанти реалізації для забезпечення найбільш коректної та оптимальної стратегії вибору, що має вплив на економічні фактори та сумісність з майбутнім програмним продуктом. Для цього застосовувався апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) передбачає собою технологію, що дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих часів, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

4.1 Постановка задачі проектування

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи прогнозу стійкості фінансових

показників. Оскільки рішення стосовно проектування та реалізації компонентів, що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу даних по компанії.

Технічні вимоги до програмного продукту є такі:

- функціонування на персональних комп'ютерах із стандартним набором компонентів;
- зручність та зрозумілість для користувача;
- швидкість обробки даних та доступ до інформації в реальному часі;
- можливість зручного масштабування та обслуговування;
- мінімальні витрати на впровадження програмного продукту.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка можливого програмного продукту, яка дозволяє аналізувати різні характеристики, що безпосередньо впливають на стійкість підприємства. Беручи за основу цю функцію, можна виділити наступні:

- F_1 – вибір моделі глибокого навчання для класифікації;
- F_2 – тип вхідних даних: з масками судин або без них;
- F_3 – спосіб постановки задачі класифікації (бінарна або багатокласова).

Кожна з цих функцій має декілька варіантів реалізації:

Функція F_1 :

- використання EfficientNetB0 як базової архітектури для класифікації;

- побудова власної простої CNN-архітектури з нуля без попередніх ваг.

Функція F_2 :

- RGB-зображення очного дна;
- бінарні маски судин;

Функція F_3 :

- бінарна класифікація;
- багатокласова класифікація за стадіями DR;

Варіанти реалізації основних функцій наведені у морфологічній карті системи на рис. 4.1.

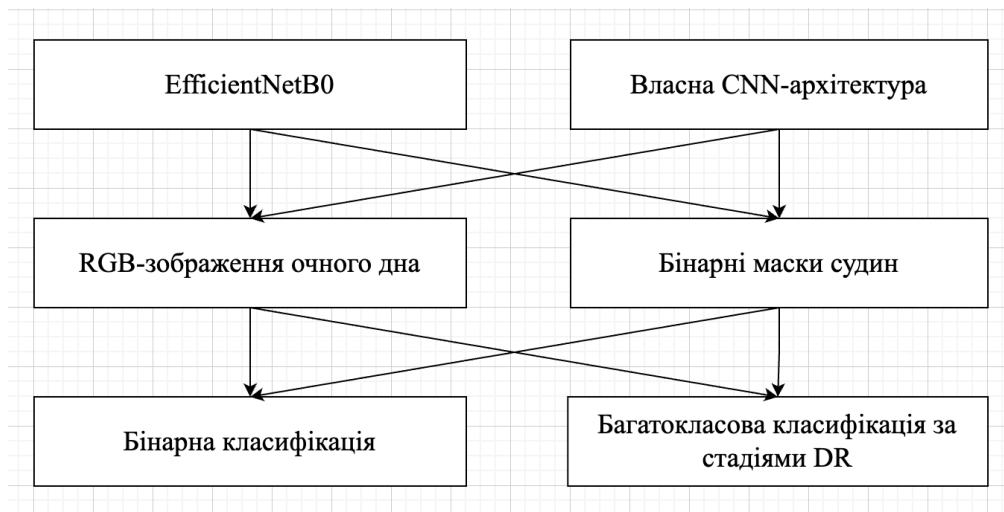


Рисунок 4.1 – Морфологічна карта системи

Морфологічна карта відображає множину всіх можливих варіантів основних функцій. Позитивно-негативна матриця показана в табл. 4.1.

Таблиця 4.1 – Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	<i>A. EfficientNetB0</i>	Висока точність, доступність попередньо навчених ваг, оптимальне співвідношення продуктивності та ресурсів	Потребує значних обчислювальних ресурсів для донавчання, складність інтерпретації
	<i>Б. Власна CNN-модель</i>	Гнучкість налаштувань, простота архітектури, легша візуалізація процесу	Нижча точність, більший ризик недонавчання, потреба в тонкому тюнінгу
F_2	<i>A. RGB-зображення очного дна</i>	Простота у використанні, збереження повної візуальної інформації, підтримка більшістю бібліотек	Надмірна інформація може «заважати» навчання, потребує попередньої обробки
	<i>Б. Маски судин</i>	Вузька фокусована увага на важливі патерни (структура судин), менший розмір зображень	Втрата колірної та текстурної інформації, залежність від якості сегментації
F_3	<i>A. Бінарна класифікація</i>	Висока чутливість, проста реалізація, ефективна для скринінгу	Не дає інформації про стадію захворювання, обмежене застосування
	<i>Б. Багатокласова класифікація</i>	Дає повну клінічну картину, корисна для формування рекомендацій	Складність моделі, потреба у збалансованому датасеті, ризик зниження recall для рідкісних класів

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізацій функцій варто відкинути, тому що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F_1 : перевагу даємо використанню EfficientNetB0 як базовій архітектурі для класифікації. Для спрощення роботи по написанню коду варіант Б має бути відкинтий.

Функція F_2 : перевагу даємо RGB-зображенням очного дна. Для спрощення роботи по написанню коду варіант Б має бути відкинтий.

Функція F_3 : програма допускає обрання обох варіантів. Можливо використати варіанти А чи Б.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

$$F_{1A} - F_{2A} - F_{3A'}$$

$$F_{1A} - F_{2A} - F_{3B'}$$

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати такі параметри:

- X_1 – середній час інференсу одного зображення (мс);

- X_2 – об’єм оперативної пам’яті, необхідної для запуску моделі (Gb);
- X_3 – час попередньої обробки зображення перед подачею в модель (мс);
- X_4 – кількість необхідних епох до досягнення прийнятної точності (шт.).

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту, як показано у табл. 4.2.

Таблиця 4.2 – Основні параметри програмного продукту

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Середній час інференсу одного зображення	X_1	мс	450	250	100
Об’єм пам’яті	X_2	Gb	10	6	3
Час попередньої обробки зображення	X_3	мс	800	400	150
Кількість епох до досягнення прийнятної точності	X_4	кількість епох	30	15	10

За даними табл. 4.2 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5.

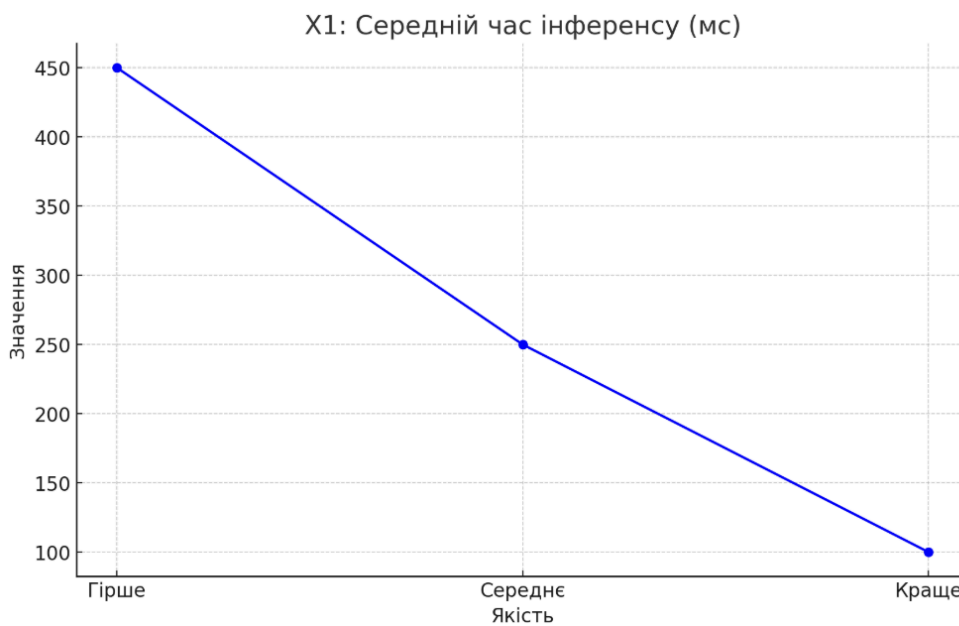


Рисунок 4.2 – X_1 , середній час інференсу одного зображення (мс)

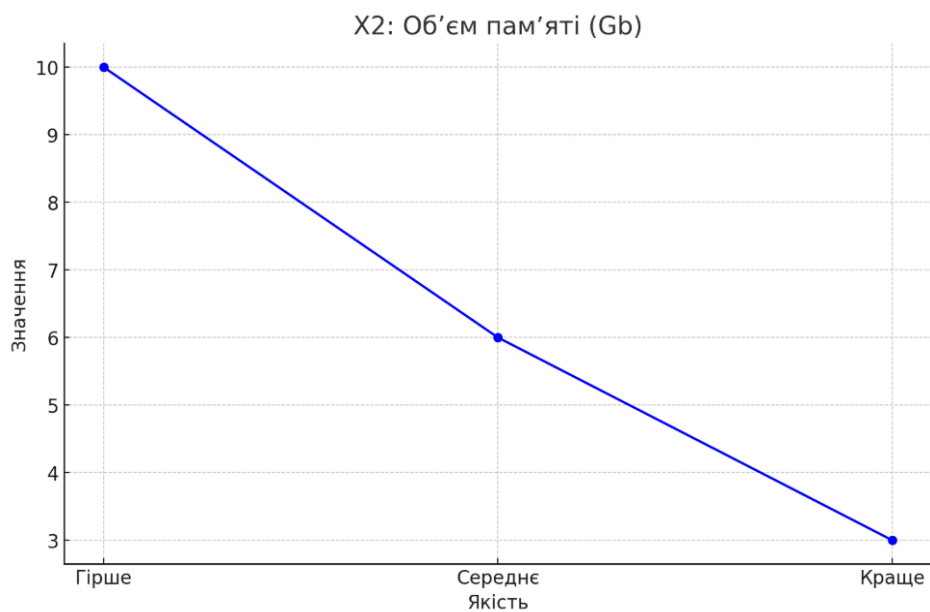


Рисунок 4.3 – X_2 , об'єм оперативної пам'яті (Gb)

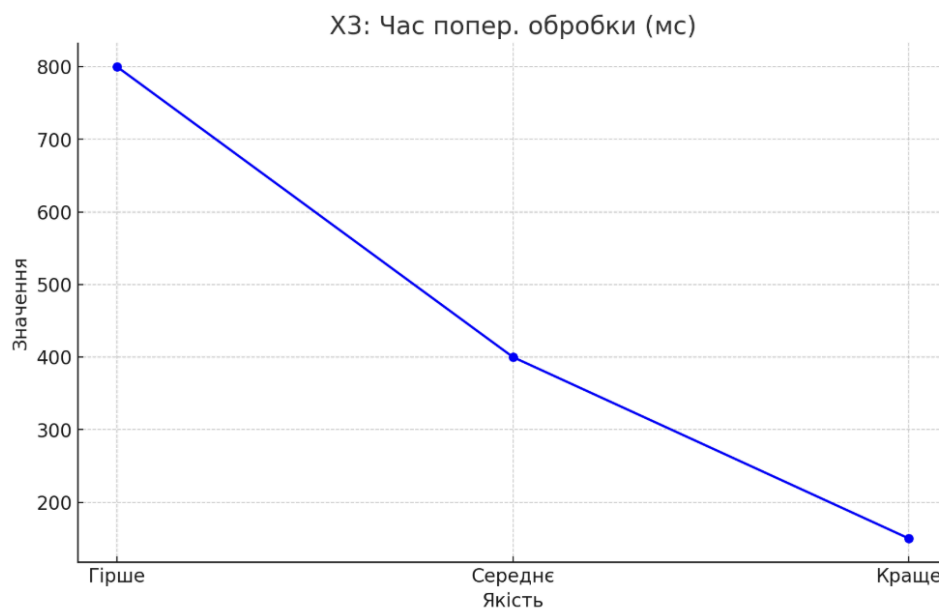


Рисунок 4.4 – X_3 , час попередньої обробки зображення (мс)

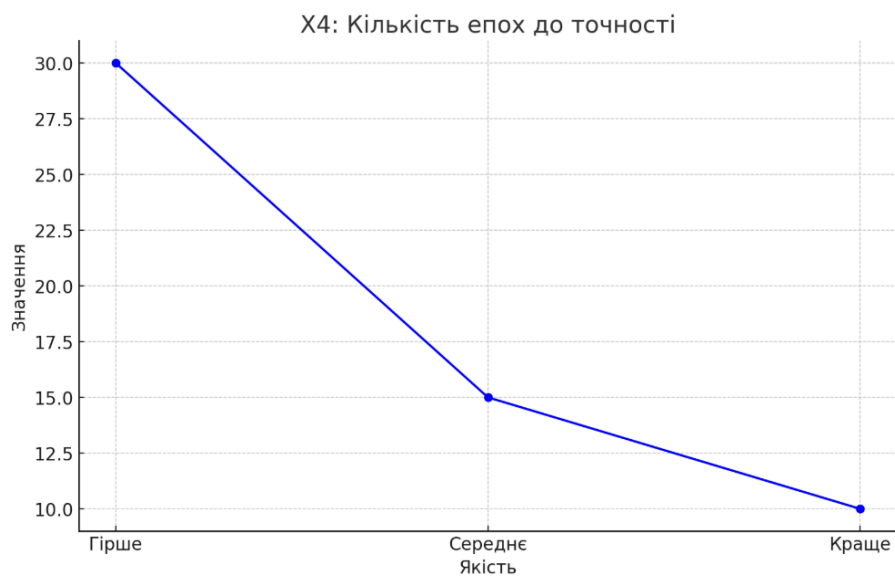


Рисунок 4.5 – X_4 , кількість необхідних епох (шт.)

4.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при прогнозуванні навантаження на CPU.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у табл. 4.3.

Таблиця 4.3 – Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
X_1	Середній час інференсу одного зображення	мс	2	1	2	1	2	3	1	12	-5,5	30,25
X_2	Об'єм пам'яті	Gb	3	3	3	2	3	1	3	18	0,5	0,25

Кінець таблиці 4.3

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
X_3	Час попередньої обробки зображення	мс	1	2	1	3	1	2	2	12	-5,5	30,25
X_4	Кількість епох до досягнення прийнятної точності	кількість епох	4	4	4	4	4	4	4	28	10,5	110,25
	Разом		10	10	10	10	10	10	10	70	0	171

Для перевірки степені достовірності експертних оцінок, визначимо такі параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70,$$

де N – число експертів;

n – кількість параметрів;

R_{ij} – ранг i -го параметру проставлений j -им експертом.

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,5.$$

Кінець таблиці 4.4

X_3 і X_4	<	<	<	<	<	<	<	<	0,5
---------------	---	---	---	---	---	---	---	---	-----

Числове значення, що визначає ступінь переваги i -го параметра над j -им, a_{ij} визначається по формулі:

$$a_{ij} = \{1.5 \text{ при } X_i > X_j; 1.0 \text{ при } X_i = X_j; 0.5 \text{ при } X_i < X_j\}.$$

З отриманих числових оцінок переваги складемо матрицю $A = \|a_{ij}\|$.

Для кожного параметра зробимо розрахунок вагомості K_{Bi} за наступними формулами:

$$K_{\text{Bi}} = \frac{b_i}{\sum_{i=1}^n b_i},$$

$$b_i = \sum_{i=1}^N a_{ij}.$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{\text{Bi}}' = \frac{b_i'}{\sum_{i=1}^n b_i'},$$

$$b_i' = \sum_{i=1}^N a_{ij} b_j'.$$

Як видно з табл. 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітерація		Друга ітерація		Третя ітерація	
	X	X	X	X	b_i	K_{Vi}	b_i^1	K_{Vi}^1	b_i^2	K_{Vi}^2
X_1	1	0,5	1,5	0,5	3,5	0,22	12,25	0,21	44,875	0,21
X_2	1,5	1	1,5	0,5	4,5	0,28	16,25	0,275	59,125	0,274
X_3	0,5	0,5	1	0,5	2,5	0,16	9,25	0,157	34,125	0,158
X_4	1,5	1,5	1,5	1	5,5	0,34	21,25	0,35	77,875	0,36
Всього:					16	1	59	1	216	1

4.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X_2 (об'єм пам'яті), X_3 (час попередньої обробки даних) та X_4 (потенційний об'єм програмного коду) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X_1 (швидкість роботи мови програмування) обрано не найгіршим.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так:

$$K_K(j) = \sum_{i=1}^n K_{Bi,j} B_{i,j},$$

де n – кількість параметрів;

K_{Bi} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 4.6 – Розрахунок вагомості параметрів

Функція	Варіант реалізації функції	Параметр	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F_1	<i>A</i>	X_1	250	10	0,21	2.1
F_2	<i>A</i>	X_2	6	7	0,274	1,918
F_3	<i>A</i>	X_3	150	8	0,158	1,264
	<i>B</i>	X_4	10	3	0,36	1,08

За даними з табл. 4.6 визначаємо рівень якості кожного з варіантів за формулою:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \dots + K_{\text{ТУ}}[F_{zk}].$$

Отримаємо:

$$K_{K1} = 2.1 + 1,918 + 1,264 = 5,282,$$

$$K_{K2} = 2.1 + 1,918 + 1,08 = 5,098.$$

Як видно з розрахунків, кращим є 1 варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.6 Економічний аналіз варіантів розробки програмного продукту

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

- розробка проекту програмного продукту;
- розробка програмної оболонки.

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як:

$$T_0 = T_p \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М'}$$

де T_p – трудомісткість розробки ПП;

K_{Π} – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення.

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеню новизни А та групи складності алгоритму 1, трудомісткість

дорівнює: $T_p = 28$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_{II} = 1.7$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.9$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 28 \cdot 1,7 \cdot 0,9 = 42,8 \text{ людино} - \text{днів.}$$

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б) – $T_p = 34$ людино-днів, $K_{II} = 1$, $K_{СК} = 1$, $K_{СТ} = 0.8$:

$$T_2 = 34 \cdot 1 \cdot 0,8 = 27,2 \text{ людино} - \text{днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (42.84 + 27.2 + 6.1 + 27.2) \cdot 8 = 716,54 \text{ людино} - \text{годин,}$$

$$T_{II} = (42.84 + 27.2 + 9.4 + 27.2) \cdot 8 = 743,84 \text{ людино} - \text{годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 23000 грн., один аналітик в області даних з окладом 28000 грн. Визначимо середню зарплату за годину за формулою:

$$СЧ = \frac{M}{T_m \cdot t} \text{ грн.,}$$

де M – місячний оклад працівників;

T_m – кількість робочих днів на тиждень;

t – кількість робочих годин в день.

$$CЧ = \frac{23000+23000+28000}{3 \cdot 21 \cdot 8} = 146,83 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою:

$$CЗП = C_ч \cdot T_i \cdot КД,$$

де $C_ч$ – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

$КД$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$C_{ЗП I} = 146,83 \cdot 716,54 \cdot 1,2 = 126251,48 \text{ грн,}$$

$$C_{ЗП II} = 146,83 \cdot 743,84 \cdot 1,2 = 131061,63 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$C_{ВІД I} = C_{ЗП I} \cdot 0,22 = 126251,48 \cdot 0,22 = 27775,32 \text{ грн,}$$

$$C_{ВІД II} = C_{ЗП II} \cdot 0,22 = 131061,63 \cdot 0,22 = 28833,56 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. (C_M).

Так як одна ЕОМ обслуговує одного програміста з окладом 23000 грн, з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_{\Gamma} = 12 \cdot M \cdot K_3 = 12 \cdot 23000 \cdot 0,2 = 55200 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{3\text{П}} = C_{\Gamma} \cdot (1 + K_3) = 55200 \cdot (1 + 0,2) = 66240 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{\text{ВІД}} = C_{3\text{П}} \cdot 0,22 = 66240 \cdot 0,22 = 14572,8 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 22000 грн.

$$C_A = K_{\text{ТМ}} \cdot K_A \cdot \text{Ц}_{\text{ПР}} = 1,4 \cdot 0,25 \cdot 22000 = 7700 \text{ грн,}$$

де $K_{\text{ТМ}}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

K_A – річна норма амортизації;

$\text{Ц}_{\text{ПР}}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{\text{ТМ}} \cdot \text{Ц}_{\text{ПР}} \cdot K_P = 1,4 \cdot 22000 \cdot 0,08 = 2464 \text{ грн,}$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{\text{ЕФ}} = (D_{\text{К}} - D_{\text{В}} - D_{\text{С}} - D_{\text{Р}}) \cdot t_3 \cdot K_{\text{В}} = 240 \cdot 8 \cdot 0,35 = 672 \text{ год},$$

де $D_{\text{К}}$ – календарна кількість днів у році;

$D_{\text{В}}, D_{\text{С}}$ – відповідно кількість вихідних та святкових днів;

$D_{\text{Р}}$ – кількість днів планових ремонтів устаткування;

t_3 – кількість робочих годин в день;

$K_{\text{В}}$ – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_{\text{С}} \cdot K_3 \cdot \text{Ц}_{\text{ЕН}} = 672 \cdot 0,2 \cdot 39,9,43 = 494,28 \text{ грн},$$

де $N_{\text{С}}$ – середньо-споживча потужність приладу;

K_3 – коефіцієнтом зайнятості приладу;

$\text{Ц}_{\text{ЕН}}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_{\text{Н}} = \text{Ц}_{\text{ПР}} \cdot 0,67 = 22000 \cdot 0,67 = 14740 \text{ грн}.$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}} \text{ грн},$$

$$C_{\text{ЕКС}} = 66240 + 14572,8 + 7700 + 2464 + 494,28 + 14740 = 106211,08 \text{ грн}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 106211,08 / 672 = 158,05 \text{ грн/год}.$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою

програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_M = C_{M-\Gamma} \cdot T,$$

$$C_{MI} = 158.04 \cdot 716.54 = 113241,98 \text{ грн},$$

$$C_{MII} = 158.04 \cdot 743.84 = 117556,47 \text{ грн}.$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{зп} \cdot 0,67,$$

$$C_{HI} = 113241,98 \cdot 0,67 = 75872,13 \text{ грн},$$

$$C_{HII} = 117556,47 \cdot 0,67 = 78762,83 \text{ грн}.$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{зп} + C_{вд} + C_M + C_H,$$

$$C_{ППI} = 126251,48 + 27775,32 + 113241,98 + 75872,13 = 343140,91 \text{ грн},$$

$$C_{ППII} = 131061,63 + 28833,56 + 113241,98 + 78762,8 = 351899,97 \text{ грн}.$$

4.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{TEPj} = K_{Kj} / C_{\Phi j},$$

$$K_{TEPj} = 5,282 / 343140,91 = 1,539 \cdot 10^{-5},$$

$$K_{TEPj} = 5,098 / 351899,97 = 1,449 \cdot 10^{-5}.$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{ТЕР1}} = 1,539 \cdot 10^{-5}$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишилися після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $K_{\text{ТЕР1}} = 1,539 \cdot 10^{-5}$.

Цей варіант реалізації програмного продукту має такі параметри:

- використання EfficientNetB0 як базової архітектури для класифікації;
- RGB-зображення очного дна;
- бінарна класифікація.

Цей варіант виконання програмного комплексу дає користувачу зручний інтерфейс, швидку реалізацію програми та доступний функціонал для роботи.

Висновки до розділу 4

В цьому розділі було проведено повний функціонально-вартісний аналіз програмного продукту. Також знайдено оцінку основних функцій програмного продукту.

На першому етапі аналізу розглянуто можливі варіанти створення системи. Проведено порівняння альтернативних рішень, розраховано коефіцієнт технічного рівня, що дало змогу обґрунтовано обрати найбільш доцільну технічну альтернативу.

У процесі аналізу визначено основні функції програмного продукту та виконано їхню оцінку. Також встановлено ключові параметри, які

характеризують функціональні можливості системи.

Друга частина аналізу була присвячена вартісній оцінці реалізації програмного продукту. Було враховано основні статті витрат – заробітну плату персоналу, споживання електроенергії, витрати на експлуатацію обчислювальної техніки тощо. За результатами розрахунків обрано оптимальний варіант реалізації, що забезпечує найкраще співвідношення між функціональністю та витратами.

Таким чином, на основі проведеного функціонально-вартісного аналізу визначено раціональний варіант розробки програмного продукту, який є технічно обґрунтованим та економічно доцільним.

ВИСНОВКИ

У процесі виконання дипломної роботи розглянуто одну з актуальних проблем сучасної медицини, а саме – автоматизоване виявлення діабетичної ретинопатії за допомогою методів глибокого навчання. Сучасна офтальмологічна практика потребує інструментів, здатних забезпечити ранню діагностику патологій сітківки, і саме тому використання штучного інтелекту в цьому контексті є виправданим як з практичної, так і з наукової точки зору.

Проведено огляд клінічних аспектів захворювання, структури доступних медичних датасетів та сучасних технологій, які застосовуються для обробки зображень очного дна. Визначено ключові особливості візуальних даних, які ускладнюють процес моделювання, серед них, зокрема, високий ступінь подібності між окремими стадіями ретинопатії, обмежений обсяг якісно розмічених даних та значний класовий дисбаланс.

У рамках дослідження реалізовано кілька варіантів моделей, серед яких: багатокласовий підхід до класифікації п'яти рівнів діабетичної ретинопатії, а також два бінарні варіанти постановки задачі, кожен з яких по-своєму розширює можливості моделі в контексті виявлення патології. Окремо було розглянуто вплив попередньої обробки зображень, зокрема, використання зображень з підсиленням судинним контуром, що дозволяє моделі зосередитись на важливих діагностичних структурах. Результати, отримані в експериментах, свідчать про те, що використання вхідних даних різного типу має помітний вплив на якість прогнозування.

Порівняння моделей за основними метриками показало, що найкращі результати досягаються при спрощенні постановки задачі (перетворенні її на бінарну класифікацію). Водночас, багатокласові підходи вимагають подальшої оптимізації, зокрема шляхом розширення вибірки. Також виявлено, що простіші за архітектурою моделі можуть бути конкурентоспроможними за умови якісної інженерії ознак чи попередньої обробки.

Окрім технічного аналізу, здійснено функціонально-вартісне оцінювання розробки, що дало змогу порівняти декілька варіантів реалізації з позиції трудозатрат і собівартості. Економічні розрахунки продемонстрували, що застосування вже існуючих, добре оптимізованих архітектур є більш доцільним у рамках обмежених ресурсів, тоді як повна розробка моделі "з нуля" вимагає значно більших витрат.

Таким чином, результати дипломної роботи підтверджують доцільність використання технологій глибокого навчання для задач первинного виявлення діабетичної ретинопатії, а також окреслюють перспективи подальшого вдосконалення моделей як з боку якості даних, так і з боку архітектурних рішень. Запропонований підхід може бути адаптований для практичного застосування в системах попереднього скринінгу та підтримки медичних рішень.

ПЕРЕЛІК ПОСИЛАНЬ

1. American Academy of Ophthalmology. Diabetic Retinopathy Preferred Practice Pattern® 2019. URL: <https://www.aao.org> (дата звернення: 12.05.2025).
2. Wilkinson C.P., Ferris F.L. III, et al. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. *Ophthalmology*. 2003;110(9):1677–1682. URL: [https://doi.org/10.1016/S0161-6420\(03\)00475-5](https://doi.org/10.1016/S0161-6420(03)00475-5) (дата звернення: 12.05.2025).
3. Gulshan V., Peng L., Coram M., et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*. 2016;316(22):2402–2410. URL: <https://jamanetwork.com/journals/jama/fullarticle/2588763> (дата звернення: 12.05.2025).
4. UK Prospective Diabetes Study (UKPDS) Group. Intensive blood-glucose control with sulphonylureas or insulin compared with conventional treatment and risk of complications in patients with type 2 diabetes. *Lancet*. 1998;352(9131):837–853. URL: <https://pubmed.ncbi.nlm.nih.gov/9742976/> (дата звернення: 12.05.2025).
5. American Academy of Ophthalmology. Diabetic Retinopathy Preferred Practice Pattern® 2019. URL: <https://www.aao.org> (дата звернення: 12.05.2025).
6. Сучасна офтальмологія – методи діагностики та лікування. [Електронний ресурс]. – URL: <https://okoplus.ua/> (дата звернення: 30.05.2025).
7. Wilkinson C.P., Ferris F.L. III, et al. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. *Ophthalmology*. 2003;110(9):1677–1682. URL: [https://doi.org/10.1016/S0161-6420\(03\)00475-5](https://doi.org/10.1016/S0161-6420(03)00475-5) (дата звернення: 12.05.2025).

8. Bhaskaranand M., Ramachandra C., Bhat S., et al. The value of automated diabetic retinopathy screening with the EyeArt system: a study of more than 100,000 consecutive encounters. *Telemedicine and e-Health*. 2019;25(9):879–885. URL: <https://doi.org/10.1089/tmj.2018.0192> (дата звернення: 12.05.2025).
9. Early Treatment Diabetic Retinopathy Study Research Group. Grading diabetic retinopathy from stereoscopic color fundus photographs – an extension of the modified Airlie House classification. *Ophthalmology*. 1991;98(5 Suppl):786–806. URL: [https://doi.org/10.1016/S0161-6420\(13\)38012-9](https://doi.org/10.1016/S0161-6420(13)38012-9) (дата звернення: 12.05.2025).
10. Gulshan V., Peng L., Coram M., et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs // *JAMA*. 2016; 316(22): 2402–2410. DOI: 10.1001/jama.2016.17216 (дата звернення: 13.05.2025).
11. U.S. Food & Drug Administration. FDA permits marketing of artificial intelligence-based device to detect certain diabetes-related eye problems. 2018. URL: <https://www.fda.gov> (дата звернення: 12.05.2025).
12. Kaggle. Diabetic Retinopathy Detection competition. 2015. URL: <https://www.kaggle.com/competitions/diabetic-retinopathy-detection> (дата звернення: 12.05.2025).
13. Lam C., Yu C., Huang L., Wong T.Y. Integrating Clinical and Imaging Data for Improved Diabetic Retinopathy Classification Using Deep Learning // *Scientific Reports*. 2022; 12: Article number: 1892. DOI: 10.1038/s41598-022-05894-5 (дата звернення: 13.05.2025).
14. Fu H., Cheng J., Xu Y., Zhang C., et al. Disc-aware ensemble network for glaucoma screening from fundus image // *IEEE Transactions on Medical Imaging*. 2018; 37(11): 2493–2501. DOI: 10.1109/TMI.2018.2837021 (дата звернення: 13.05.2025).
15. Kaggle. Diabetic Retinopathy Detection Challenge. <https://www.kaggle.com/competitions/diabetic-retinopathy-detection>

16. Li, Z. et al. (2019). Deep learning for detecting retinal detachment and discerning macular status using ultra-widefield fundus images. *Communications Biology*, 2(1), 1–8. Dataset: <https://github.com/uw-biomedical-ml/DDR-dataset>
17. Kumar, S. (2022). Diabetic Retinopathy Dataset. <https://www.kaggle.com/datasets/sachinkumar413/diabetic-retinopathy-dataset>
18. Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229. (дата звернення: 14.05.2025).
19. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260. (дата звернення: 14.05.2025).
20. Gulshan, V., Peng, L., Coram, M., et al. (2016). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy. *JAMA*, 316(22), 2402–2410. (дата звернення: 14.05.2025).
21. Esteva, A., Robicquet, A., Ramsundar, B., et al. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24–29. (дата звернення: 14.05.2025).
22. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. (дата звернення 14.05.2025)
23. Rawat, W., & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9), 2352–2449. (дата звернення 14.05.2025)
24. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *International Conference on Machine Learning (ICML)*. (дата звернення 14.05.2025)
25. Glorot, X., & Bengio, Y. (2010). *Understanding the difficulty of training deep feedforward neural networks*. Proceedings of the Thirteenth International

- Conference on Artificial Intelligence and Statistics (AISTATS). (дата звернення 15.05.2025)
26. Nair, V., & Hinton, G. E. (2010). *Rectified Linear Units improve Restricted Boltzmann Machines*. Proceedings of the 27th International Conference on Machine Learning (ICML). (дата звернення 15.05.2025)
 27. Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). *Focal Loss for Dense Object Detection*. Proceedings of the IEEE International Conference on Computer Vision (ICCV). (дата звернення 15.05.2025)
 28. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). *Dermatologist-level classification of skin cancer with deep neural networks*. *Nature*, 542(7639), 115–118. (дата звернення 15.05.2025)
 29. Wang, G., Ye, J.C., Mueller, K., Fessler, J.A. “Image reconstruction is a new frontier of machine learning.” *IEEE Transactions on Medical Imaging*, 2018. (дата звернення 15.05.2025)
 30. Morley, J., Machado, C. C. V., Burr, C., Cows, J., Joshi, I., Taddeo, M., & Floridi, L. (2020). The ethics of AI in health care: A mapping review. *Social Science & Medicine*, 260, 113172. (дата звернення 16.05.2025)

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
from google.colab import drive
from os import listdir
import cv2
import os
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
from PIL import Image
from skimage.filters import frangi
from skimage.morphology import opening, disk
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import EfficientNetB3
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout,
Input, Concatenate
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import LambdaCallback, EarlyStopping,
ReduceLROnPlateau
import tensorflow.keras.layers as layers
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau,
ModelCheckpoint
from sklearn.metrics import classification_report, confusion_matrix,
ConfusionMatrixDisplay
from tensorflow.keras.metrics import Precision, Recall

drive.mount('/content/drive', force_remount=True)

dataset_path = '/content/drive/MyDrive/Colab Notebooks/binary_retina'
print("Вміст папки Diabetic_retinography_detection:", listdir(dataset_path))

number_of_classes = len(listdir(dataset_path))

classes = os.listdir(dataset_path)

image_counts = {}

for cls in classes:
    cls_path = os.path.join(dataset_path, cls)
```

```

    if os.path.isdir(cls_path):
        images = os.listdir(cls_path)
        image_counts[cls] = len(images)

print(f"Кількість класів: {number_of_classes}")
print("Кількість зображень на клас:")
print(image_counts)

plt.bar(image_counts.keys(), image_counts.values())
plt.title("Кількість зображень на клас")
plt.xlabel("Класи")
plt.ylabel("Кількість зображень")
plt.xticks(rotation=45)
plt.show()

import os
import shutil
import random

input_path = '/content/drive/MyDrive/Colab Notebooks/binary_retina'
output_path = '/content/drive/MyDrive/Colab Notebooks/binary_retina_split'
split_ratios = {'train': 0.9, 'val': 0.05, 'test': 0.05}

random.seed(42)

if os.path.exists(output_path):
    shutil.rmtree(output_path)
os.makedirs(output_path)

for cls in os.listdir(input_path):
    cls_path = os.path.join(input_path, cls)
    if not os.path.isdir(cls_path):
        continue

    images = os.listdir(cls_path)
    random.shuffle(images)

    total = len(images)
    n_train = int(total * split_ratios['train'])
    n_val = int(total * split_ratios['val'])
    n_test = total - n_train - n_val

    split_data = {
        'train': images[:n_train],

```

```

        'val': images[n_train:n_train + n_val],
        'test': images[n_train + n_val:]
    }

    for split, imgs in split_data.items():
        for img in imgs:
            src = os.path.join(cls_path, img)
            dst_dir = os.path.join(output_path, split, cls)
            dst = os.path.join(dst_dir, img)

            os.makedirs(dst_dir, exist_ok=True)

            if os.path.exists(src):
                shutil.copyfile(src, dst)
            else:
                print(f"Пропущено: {src}")

print("Розбиття завершено!")

# Багатокласова модель для прогнозування стадії діабетичної ретинопатії чи
відсутності ретинопатії

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    brightness_range=[0.8, 1.2],
    shear_range=0.1
)

val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Colab Notebooks/binary_retina_split/train',
    target_size=(224, 224),
    color_mode='rgb',
    batch_size=32,
    class_mode='categorical',
    shuffle=True,

```

```

    seed=42
)

val_generator = val_datagen.flow_from_directory(
    '/content/drive/MyDrive/Colab Notebooks/binary_retina_split/val',
    target_size=(224, 224),
    color_mode='rgb',
    batch_size=32,
    class_mode='categorical',
    shuffle=False,
    seed=42
)

test_datagen = ImageDataGenerator(rescale=1./255)

test_generator = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/Colab Notebooks/binary_retina_split/test',
    target_size=(224, 224),
    color_mode='rgb',
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)

from tensorflow.keras import layers, models

model = models.Sequential([
    layers.Input(shape=(224, 224, 3)),

    layers.Conv2D(32, (3, 3), activation='relu'),
    layers.MaxPooling2D(2, 2),

    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D(2, 2),

    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D(2, 2),

    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(5, activation='softmax') # 5 класів
])

```

```

])

model.summary()

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

from sklearn.utils.class_weight import compute_class_weight
import numpy as np
from tensorflow.keras.metrics import Precision, Recall

class_labels = list(train_generator.class_indices.keys())
class_indices = train_generator.class_indices

image_counts = {
    'Healthy': 1000,
    'Moderate DR': 900,
    'Mild DR': 370,
    'Proliferate DR': 290,
    'Severe DR': 190
}

y_train = []
for cls in class_labels:
    y_train.extend([class_indices[cls]] * image_counts[cls])

class_weights_array = compute_class_weight(
    class_weight='balanced',
    classes=np.unique(y_train),
    y=y_train
)

class_weights = dict(enumerate(class_weights_array))
print("Class weights:", class_weights)

model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy', Precision(name='precision'), Recall(name='recall')]
)

history = model.fit(

```

```

train_generator,
validation_data=val_generator,
epochs=20,
class_weight=class_weights
)

model.save('/content/drive/MyDrive/Colab Notebooks/simple_cnn_retina.h5')

from tensorflow.keras.preprocessing.image import ImageDataGenerator

test_datagen = ImageDataGenerator(rescale=1./255)

test_generator = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/Colab Notebooks/binary_retina_split/test',
    target_size=(224, 224),
    color_mode='rgb',
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)

test_loss, test_acc, test_precision, test_recall =
model.evaluate(test_generator)

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix

class_labels = list(test_generator.class_indices.keys())

y_pred_probs = model.predict(test_generator)
y_pred = np.argmax(y_pred_probs, axis=1)
y_true = test_generator.classes

print(f"Test Accuracy: {test_acc:.4f}")
print(f"Test Precision: {test_precision:.4f}")
print(f"Test Recall: {test_recall:.4f}")

print("\nClassification Report:")
print(classification_report(y_true, y_pred, target_names=class_labels))

cm = confusion_matrix(y_true, y_pred)

plt.figure(figsize=(8,6))

```

```

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
             xticklabels=class_labels, yticklabels=class_labels)
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Confusion Matrix on Test Data')
plt.tight_layout()
plt.show()

# Бінарна модель для виявлення діабетичної ретинопатії

from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMG_SIZE = 299
BATCH_SIZE = 32

datagen = ImageDataGenerator(rescale=1./255)

train_path = "/content/drive/MyDrive/Colab
Notebooks/binary_retina_balanced/train"
valid_path = "/content/drive/MyDrive/Colab
Notebooks/binary_retina_balanced/valid"
test_path = "/content/drive/MyDrive/Colab
Notebooks/binary_retina_balanced/test"

train_gen = datagen.flow_from_directory(
    train_path,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    shuffle=True
)

val_gen = datagen.flow_from_directory(
    valid_path,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    shuffle=False
)

test_gen = datagen.flow_from_directory(
    test_path,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,

```

```

class_mode='binary',
shuffle=False
)

import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.preprocessing.image import ImageDataGenerator

modell = models.Sequential([
    layers.Input(shape=(IMG_SIZE, IMG_SIZE, 3)),

    layers.Conv2D(32, (3, 3), activation='relu'),
    layers.MaxPooling2D(2, 2),

    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D(2, 2),

    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D(2, 2),

    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(1, activation='sigmoid')
])

modell.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy', 'precision', 'recall']
)

history1 = modell.fit(
    train_gen,
    validation_data=val_gen,
    epochs=10
)

test_loss, test_acc, test_prec, test_recall = modell.evaluate(test_gen)
print(f"Test loss: {test_loss:.4f}")
print(f"Test accuracy: {test_acc:.4f}")
print(f"Test precision: {test_prec:.4f}")
print(f"Test recall: {test_recall:.4f}")

```

```

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.legend()
plt.title('Accuracy')

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.legend()
plt.title('Loss')

plt.show()

# Бінарна модель для визначення 1 стадії ДР

import os
import shutil
import random
from pathlib import Path

ORIG_PATH = '/content/drive/MyDrive/Colab Notebooks/binary_retina'
DEST_PATH = "/content/drive/MyDrive/Colab Notebooks/binary_mild_vs_rest"

TARGET_MILD = 370
TARGET_OTHER = 370

def split_counts(total):
    train = int(total * 0.7)
    val = int(total * 0.15)
    test = total - train - val
    return train, val, test

def copy_subset(images, counts, dest_base, label):
    split_names = ['train', 'val', 'test']
    start = 0
    for i, count in enumerate(counts):
        subset = images[start:start+count]
        target_dir = os.path.join(dest_base, split_names[i], label)
        os.makedirs(target_dir, exist_ok=True)
        for img_path in subset:

```

```

        shutil.copy(img_path, os.path.join(target_dir,
os.path.basename(img_path)))
        start += count

mild_images = list(Path(os.path.join(ORIG_PATH, "Mild DR")).glob("*"))
random.shuffle(mild_images)
mild_images = mild_images[:TARGET_MILD]
mild_counts = split_counts(len(mild_images))
copy_subset(mild_images, mild_counts, DEST_PATH, "Mild DR")

other_images = []
for cls in ["Moderate DR", "Severe DR", "Proliferate DR"]:
    cls_path = os.path.join(ORIG_PATH, cls)
    images = list(Path(cls_path).glob("*"))
    random.shuffle(images)
    take = TARGET_OTHER // 3
    other_images.extend(images[:take])

random.shuffle(other_images)
other_counts = split_counts(len(other_images))
copy_subset(other_images, other_counts, DEST_PATH, "other")

print("Датасет збережено в:", DEST_PATH)

binary_train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    '/content/drive/MyDrive/Colab Notebooks/binary_mild_vs_rest/train',
    label_mode='categorical',
    image_size=(299, 299),
    batch_size=32,
    shuffle=True
)

binary_val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    '/content/drive/MyDrive/Colab Notebooks/binary_mild_vs_rest/val',
    label_mode='categorical',
    image_size=(299, 299),
    batch_size=32,
    shuffle=False
)

binary_test_ds = tf.keras.preprocessing.image_dataset_from_directory(
    '/content/drive/MyDrive/Colab Notebooks/binary_mild_vs_rest/test',
    label_mode='categorical',

```

```

    image_size=(299, 299),
    batch_size=32,
    shuffle=False
)

AUTOTUNE = tf.data.AUTOTUNE
binary_train_ds = binary_train_ds.prefetch(buffer_size=AUTOTUNE)
binary_val_ds = binary_val_ds.prefetch(buffer_size=AUTOTUNE)
binary_test_ds = binary_test_ds.prefetch(buffer_size=AUTOTUNE)

efficientnet_base = EfficientNetB0(
    input_shape=(IMG_SIZE, IMG_SIZE, 3),
    include_top=False,
    weights='imagenet'
)
efficientnet_base.trainable = False

efficientnet_model = models.Sequential([
    efficientnet_base,
    layers.GlobalAveragePooling2D(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.4),
    layers.Dense(NUM_CLASSES, activation='softmax')
])

efficientnet_model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
    loss='categorical_crossentropy',
    metrics=['accuracy', Precision(name='precision'), Recall(name='recall')]
)

history_efficientnet = efficientnet_model.fit(
    binary_train_ds,
    validation_data=binary_val_ds,
    epochs=EPOCHS
)

def evaluate_model(model, test_ds, class_names):
    y_true = []
    y_pred = []

    for images, labels in test_ds:
        preds = model.predict(images)
        y_true.extend(np.argmax(labels.numpy(), axis=1))

```

```

        y_pred.extend(np.argmax(preds, axis=1))

print("📄 Classification Report:")
print(classification_report(y_true, y_pred, target_names=class_names))

cm = confusion_matrix(y_true, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=class_names)
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix")
plt.show()

class_names = ['Mild DR', 'Other']

print("🔍 EfficientNetB0 Evaluation")
evaluate_model(efficientnet_model, binary_test_ds, class_names)

efficientnet_base.trainable = True
for layer in efficientnet_base.layers[:-20]:
    layer.trainable = False

efficientnet_model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-5),
    loss='categorical_crossentropy',
    metrics=['accuracy', Precision(name='precision'), Recall(name='recall')]
)

history_finetuned = efficientnet_model.fit(
    binary_train_ds,
    validation_data=binary_val_ds,
    epochs=8
)

test_loss, test_acc, test_prec, test_rec =
efficientnet_model.evaluate(binary_test_ds)
print(f"\n🔍 Test Accuracy: {test_acc:.4f}")
print(f"🔍 Test Precision: {test_prec:.4f}")
print(f"🔍 Test Recall: {test_rec:.4f}")

class_names = ['Mild DR', 'other']

import seaborn as sns
y_true = []
y_pred = []

```

```

for images, labels in binary_test_ds:
    preds = efficientnet_model.predict(images)
    y_true.extend(np.argmax(labels.numpy(), axis=1))
    y_pred.extend(np.argmax(preds, axis=1))

print("\n📄 Classification Report:")
print(classification_report(y_true, y_pred, target_names=class_names))

cm = confusion_matrix(y_true, y_pred)
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=class_names,
yticklabels=class_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

target_class_name = 'other'
target_class_index = class_names.index(target_class_name)

for images, labels in binary_test_ds.take(2):
    for i in range(len(images)):
        label = labels[i].numpy()
        true_index = np.argmax(label)

        if true_index == target_class_index:
            image = images[i]
            true_class = class_names[true_index]

            img_array = tf.expand_dims(image, axis=0)
            prediction = efficientnet_model.predict(img_array)
            predicted_index = np.argmax(prediction)
            predicted_class = class_names[predicted_index]

            print(f"\n🟢 Ground truth: {true_class}")
            print(f"\n🟡 Model prediction: {predicted_class}")
            print(f"\n📊 Probabilities: {dict(zip(class_names,
np.round(prediction[0], 2)))}")

            plt.imshow(image.numpy().astype("uint8"))
            plt.axis("off")
            plt.title(f"Predicted: {predicted_class} | True: {true_class}")
            plt.show()

```

```

        break

import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf

for images, labels in binary_test_ds.take(1):
    for i in range(2):
        image = images[i]
        label = labels[i]

        img_array = tf.expand_dims(image, axis=0)
        prediction = efficientnet_model.predict(img_array)
        predicted_index = np.argmax(prediction)
        predicted_class = class_names[predicted_index]

        true_index = np.argmax(label)
        true_class = class_names[true_index]

        print(f"\n🟢 Ground truth: {true_class}")
        print(f"🟡 Model prediction: {predicted_class}")
        print(f"📊 Probabilities: {dict(zip(class_names,
np.round(prediction[0], 2)))}")

        plt.imshow(image.numpy().astype("uint8"))
        plt.axis("off")
        plt.title(f"Predicted: {predicted_class} | True: {true_class}")
        plt.show()

# Використання багатоступенчастої моделі класифікації

import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image

IMG_SIZE = 299
class_names_stage1 = ['Diseased', 'Healthy']
class_names_stage2 = ['Mild DR', 'other']

def preprocess_image(img_path):
    img = image.load_img(img_path, target_size=(IMG_SIZE, IMG_SIZE))

```

```

img_array = image.img_to_array(img)
img_array = img_array / 255.0
img_array = np.expand_dims(img_array, axis=0)
return img_array

def predict_image(img_path):
    img_array = preprocess_image(img_path)

    # Етап 1: первинна модель
    pred1 = model1.predict(img_array)[0][0]
    class_idx1 = int(pred1 > 0.5)
    class_name1 = class_names_stage1[class_idx1]

    print(f"\n🔍 Stage 1 prediction: {class_name1} ({{pred1:.2f}}
probability)")

    if class_name1 == 'Healthy':
        print("✅ Висновок: пацієнт здоровий")
        class_name2 = None
    else:
        # Етап 2: уточнення ступеня захворювання
        pred2 = efficientnet_model.predict(img_array)[0]
        class_idx2 = np.argmax(pred2)
        class_name2 = class_names_stage2[class_idx2]

        print(f"Висновок: хворий ({{class_name2}})")
        print(f"Ймовірності: {dict(zip(class_names_stage2, np.round(pred2,
2)))}")

    plt.imshow(tf.keras.utils.load_img(img_path))
    plt.axis('off')
    plt.title(f"Stage 1: {class_name1}" + (f" → Stage 2: {class_name2}" if
class_name1 == 'Diseased' else ""))
    plt.show()

predict_image('/content/drive/MyDrive/Colab
Notebooks/binary_retina_balanced/test/Disease/Proliferate_DR_Proliferate
DR_146.png')

# Експеримент з використанням масок судин

import cv2
from skimage.filters import frangi
from skimage.morphology import opening, disk

```

```

input_folder = "/content/drive/MyDrive/Colab
Notebooks/binary_retina_balanced/train/Healthy"
output_folder = '/content/drive/MyDrive/Colab
Notebooks/binary_retina_masks1/train/Healthy'

def process_image(image_path, output_path):
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    if image is None:
        print(f"Помилка: не вдалося завантажити {image_path}")
        return

    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    image_enhanced = clahe.apply(image)

    blurred = cv2.GaussianBlur(image_enhanced, (3, 3), 0)

    vessel_mask = frangi(blurred, sigmas=np.linspace(0.5, 3, 3))

    vessel_mask[vessel_mask >= 0.5] = 255
    vessel_mask[vessel_mask < 0.5] = 0

    vessel_mask = opening(vessel_mask, disk(1))

    cv2.imwrite(output_path, vessel_mask)
    print(f"Збережено: {output_path}")

for filename in os.listdir(input_folder):
    if filename.lower().endswith(('.png', '.jpg', '.jpeg')):
        input_path = os.path.join(input_folder, filename)
        output_path = os.path.join(output_folder, 'Mask_' + filename)
        process_image(input_path, output_path)

print("Всі маски збережено!")

from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMG_SIZE = 299
BATCH_SIZE = 32

datagen = ImageDataGenerator(rescale=1./255)

train_path = "/content/drive/MyDrive/Colab
Notebooks/binary_retina_masks1/train"

```

```

valid_path = "/content/drive/MyDrive/Colab
Notebooks/binary_retina_masks1/valid"
test_path  = "/content/drive/MyDrive/Colab
Notebooks/binary_retina_masks1/test"

train_gen = datagen.flow_from_directory(
    train_path,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    shuffle=True
)

val_gen = datagen.flow_from_directory(
    valid_path,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    shuffle=False
)

test_gen = datagen.flow_from_directory(
    test_path,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    shuffle=False
)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
Dropout

model = Sequential([
    Conv2D(8, (3,3), activation='relu', input_shape=(299, 299, 3)),
    MaxPooling2D(2,2),

    Conv2D(16, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dense(16, activation='relu'),
    Dense(1, activation='sigmoid')
])

```

```

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

model.summary()

history = model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=10
)

test_loss, test_acc, test_prec, test_recall = model.evaluate(test_gen)
print(f"Test loss: {test_loss:.4f}")
print(f"Test accuracy: {test_acc:.4f}")
print(f"Test precision: {test_prec:.4f}")
print(f"Test recall: {test_recall:.4f}")

import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report,
ConfusionMatrixDisplay

y_true = test_gen.classes
y_pred_prob = model.predict(test_gen)
y_pred = (y_pred_prob > 0.5).astype(int).flatten()

cm = confusion_matrix(y_true, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=test_gen.class_indices.keys())

plt.figure(figsize=(6, 6))
disp.plot(cmap=plt.cm.Blues, values_format='d')
plt.title("Матриця плутанини")
plt.show()

print("🔍 Класифікаційний звіт:")
print(classification_report(y_true, y_pred,
target_names=test_gen.class_indices.keys()))

```