

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

ШУЛЬГА МАКСИМ ВОЛОДИМИРОВИЧ

УДК 004.032.26 (043.3)

ДИСЕРТАЦІЯ

МЕТОД ТА ПРОГРАМНІ ЗАСОБИ МУЛЬТИМОДАЛЬНОГО
АНАЛІЗУ МЕДИЧНИХ ДАНИХ НА ОСНОВІ ГЛИБОКОГО
НАВЧАННЯ

121 Інженерія програмного забезпечення

12 Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело _____

Науковий керівник: Гордієнко Юрій Григорович, професор кафедри ОТ,
д.ф.-м.н., с.н.с.

Київ – 2023

АНОТАЦІЯ

Шульга М.В. Метод та програмні засоби мультимодального аналізу медичних даних на основі глибокого навчання. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 121 – Інженерія програмного забезпечення з галузі знань 12 – Інформаційні технології. – Національний Технічний Університет України «Київський Політехнічний Інститут імені Ігоря Сікорського», Київ, 2023.

Дисертаційна робота присвячена розробці методу та програмних засобів мультимодального аналізу медичних даних на основі глибокого навчання, що дозволяє підвищити точність багатокласової класифікації.

Останнім часом сфера медицини активно використовує можливості штучного інтелекту. Зокрема, глибокі нейронні мережі довели свою ефективність і придатність для автоматизованого виявлення та класифікації захворювань. Інтеграція методів штучного інтелекту має потенціал для оптимізації та покращення точності програм скринінгу шляхом автоматизації аналізу медичних даних, усуваючи необхідність безпосередньої участі медичного персоналу на етапі скринінгу. Нагальною проблемою в цьому контексті є діагностика діабетичної ретинопатії, поширеного ускладнення діабету, що призводить до погіршення зору серед дорослого населення світу. Отже, в даний час існує актуальна потреба в застосуванні підходу з використанням комп'ютерного зору та глибоких нейронних мереж для надання передових медичних послуг за допомогою штучного інтелекту, з особливим акцентом на класифікації діабетичної ретинопатії.

Тема дисертаційної роботи входить в план наукової роботи затвердженому на кафедрі обчислювальної техніки КПІ ім. Ігоря Сікорського, що враховує розпорядження Кабінету Міністрів України від 2 грудня 2020 р. № 1556-р про схвалення Концепції розвитку штучного інтелекту в Україні.

Метою дисертаційної роботи є підвищення точності багатокласової класифікації захворювань глибокими нейронними мережами, шляхом розробки

методу мультимодального аналізу медичних даних. Об'єктом дослідження є процеси аналізу вимог, розробки, впровадження і супроводження програмного забезпечення для мультимодального аналізу медичних даних на основі глибокого навчання, які сприяють підвищенню точності визначення окремих класів для завдання багатокласової класифікації медичних даних; а предметом дослідження – методи і моделі розробки і супроводу програмного забезпечення для мультимодального аналізу медичних даних на основі глибоких нейронних мереж різної архітектури і способів організації додаткових модальностей.

Методичною основою дослідження є системне опрацювання та аналіз теоретичного матеріалу, присвяченого підвищенню точності вирішення завдання багатокласової класифікації, шляхом аналізу медичних даних за допомогою моделей глибоких нейронних мереж.

Для досягнення поставленої мети були вирішені такі завдання:

- було досліджено сучасні методи та системи виявлення та багатокласової класифікації захворювань на прикладі діабетичної ретинопатії;
- було виявлено шляхи покращення існуючих методів та систем виявлення та багатокласової класифікації захворювань на прикладі діабетичної ретинопатії;
- було розроблено метод мультимодального аналізу медичних даних на основі доповнення метаданих;
- було розвинуто спосіб підвищення точності багатокласової класифікації завдяки використанню методу мультимодального аналізу для різних архітектур згорткової компоненти нейронної мережі;
- було розроблено метод недетермінованого штучного доповнення метаданих;
- було розвинуто спосіб підвищення точності багатокласової класифікації завдяки використанню методу недетермінованого штучного доповнення метаданих для різних стандартних і спеціалізованих медичних наборів даних;
- було розроблено математичний опис для оцінки підвищення точності багатокласової класифікації на різних стандартних і спеціалізованих медичних

наборах даних.

За результатами проведеного дослідження та згідно поставленого завдання було запропоновано комплексний метод мультимодального аналізу медичних даних на основі глибокого навчання, при розробці якого були проведені наступні дослідження:

- дослідження доповнення метаданих для вирішення завдання багатокласової класифікації;
- дослідження впливу складності мультимодальної моделі на вирішення завдання багатокласової класифікації;
- дослідження недетермінованого доповнення метаданих для вирішення завдання багатокласової класифікації;
- дослідження впливу мультимодального доповнення метаданих на точність багатокласової класифікації.

Розглянуто проблему багатокласової класифікації для одномодальної (з введенням зображення) моделі та мультимодальної (з введенням зображення та тексту) моделі та створено кілька варіантів вхідних значень і відповідних моделей на основі аналізу способів та методів використання глибокого навчання для виявлення захворювань на прикладі діабетичної ретинопатії і аналізу існуючих методологій виявлення захворювань: одномодальна модель (SM) лише з вхідним зображенням і мультимодальні моделі з вхідними зображеннями та текстом, як-от мультимодальна модель із думкою пацієнта (MP), мультимодальна модель із думкою експерта (ME), мультимодальна модель із думкою пацієнта та експерта (MPE) та мультимодальна модель з недетермінованою думкою експерта (MMFE). Вплив додаткових даних, таких як суб'єктивна думка “пацієнта” про свій стан здоров'я та думка “експерта” (що забезпечує “витік даних”), може бути корисним у деяких практичних ситуаціях. Думки пацієнтів та експертів були імітовані додатковими (доповненими) даними, отриманими з змодельованих анкет.

Проведено дослідження доповнення метаданих для вирішення завдання багатокласової класифікації, яке показало, що всі створені мультимодальні моделі (MP, ME, MPE) у порівнянні з одномодальною моделлю (SM) дозволили досягти

різних статистично значущих покращень точності багатокласової класифікації за значенням площі під кривою похибок (AUC) для всіх класів у діапазоні від 4% до 27%, що виходять за межі стандартного відхилення 2-3% вимірюного перехресною перевіркою.

Проведено дослідження впливу складності мультимодальної моделі на вирішення завдання багатокласової класифікації, яке показало, що мультимодальна модель (MP) у порівнянні з одномодальною моделлю (SM) дозволила досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC для деяких класів у діапазоні від 15% до 26% (в залежності від складності архітектури згорткової компоненти нейронної мережі), що виходять за межі стандартного відхилення 3-8% вимірюного перехресною перевіркою.

Проведено дослідження недетермінованого доповнення метаданих для вирішення завдання багатокласової класифікації, яке показало, що мультимодальна модель (MMFE) у порівнянні з одномодальною моделлю (SM) дозволила досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC у діапазоні від 12% до 26%.

Наведено теоретичні оцінки можливих (найкращих – коли точність за класами підвищується на сумарну похибку точності за окремими класами для одномодальної моделі, де похибку вдається зменшити завдяки застосуванню маркування у додатковій модальності; проміжних – коли точність за класами підвищується на сумарну похибку точності, помножену на функцію розподілу ймовірностей, де продемонстровано можливу залежність покращення від середньої точності для одномодальної моделі; і найгірших – коли точність за класами не підвищується взагалі) рівнів покращення точності багатокласової класифікації за допомогою простого математичного опису з акцентом на деяких практичних випадках.

На основі теоретичних оцінок проведено дослідження впливу мультимодального доповнення метаданих на точність багатокласової класифікації з використанням різних стандартних (CIFAR10) і спеціалізованих медичних

(PathMNIST, RetinaMNIST) наборів даних, яке показало, що точність мультимодальних моделей значно змінюється залежно від складності набору даних, розміру вибірки та мінливості даних.

Запропоновано комплексний метод мультимодального аналізу медичних даних на основі глибокого навчання, який полягає у використанні “витоку даних” на крайніх і подібних класах, що дозволяє підвищити точність визначення окремих класів для вирішення завдань багатокласової класифікації.

Проведено аналіз результатів використання запропонованого комплексного методу мультимодального аналізу медичних даних, який показав, що запропонований метод на відміну від існуючих рішень за рахунок використання додаткових модальностей на основі доповнення метаданих дозволив отримати підвищення точності визначення окремих класів для завдань багатокласової класифікації захворювань на 4-27% у порівнянні зі стандартним одномодальним підходом для розглянутих ідентичних наборів даних і архітектур нейронних мереж.

Ключові слова: багатокласова класифікація, нейронні мережі, глибоке навчання, доповнення метаданих, недетерміноване маркування, мультимодальна модель, сітківка, діабетична ретинопатія.

ABSTRACT

Shulha M.V. Method and software tools of multimodal analysis of medical data based on deep learning. – Qualified scientific work on the rights of the manuscript.

Dissertation for the degree of Doctor of Philosophy in the specialty 121 – Software Engineering and 12 – Information Technologies. – National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, 2023.

Recently, the field of medicine has been actively using the capabilities of artificial intelligence. In particular, deep neural networks have proven their effectiveness and suitability for automated disease detection and classification. The integration of artificial intelligence techniques has the potential to optimize and improve the accuracy of screening programs by automating the analysis of medical data, eliminating the need for direct involvement of medical personnel at the screening stage. An urgent problem in this context is the diagnosis of diabetic retinopathy, a common complication of diabetes that leads to visual impairment among the adult population of the world. Hence, there is currently an urgent need to apply a computer vision and deep neural network approach to provide advanced healthcare services with the help of artificial intelligence, with special emphasis on the classification of diabetic retinopathy.

The topic of the dissertation is included in the plan of scientific work approved by Computer engineering department of National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, which takes into account the order of the Cabinet of Ministers of Ukraine of December 2, 2020 # 1556-r on the approval of the Concept of the Development of Artificial Intelligence in Ukraine.

The goal of the dissertation is to increase the accuracy of multiclass classification of diseases by deep neural networks, by developing a method of multimodal analysis of medical data. The object of research is the processes of analysis of requirements, development, implementation and maintenance of software for multimodal analysis of medical data based on deep learning, which contribute to increasing the accuracy of determining individual classes for the task of multi-class classification of medical data; and the subject of research is methods and models of software development and support

for multimodal analysis of medical data based on deep neural networks of various architectures and ways of organizing additional modalities.

The methodological basis of the study is the systematic processing and analysis of theoretical material devoted to increasing the accuracy of solving the problem of multiclass classification by analyzing medical data using deep neural network models.

To achieve the goal, the following tasks were solved:

- modern methods and systems of detection and multiclass classification of diseases were studied using the example of diabetic retinopathy;
- ways of improving the existing methods and systems of detection and multiclass classification of diseases using the example of diabetic retinopathy were identified;
- a method of multimodal analysis of medical data based on the metadata augmentation was developed;
- a method of increasing the accuracy of multiclass classification was developed due to the use of the multimodal analysis method for different architectures of the convolutional component of the neural network;
- a method of fuzzy metadata augmentation was developed;
- a method of increasing the accuracy of multiclass classification was developed through the use of the method of fuzzy metadata augmentation for various standard and specialized medical data sets;
- a mathematical model was developed to evaluate the accuracy improvement of multiclass classification on various standard and specialized medical datasets.

According to the results of the research and according to the task, a complex method of multimodal analysis of medical data based on deep learning was proposed, during the development of which the following research were conducted:

- research of the metadata augmentation for the multiclass classification;
- research of the impact of multimodal model complexity on multiclass classification;
- research of fuzzy metadata augmentation for the multiclass classification;
- research of the effect of multimodal metadata augmentation on multiclass

classification.

The problem of multiclass classification for a single modal (with image input) model and a multimodal (with image and text input) model was considered, and several options for input values and corresponding models were created based on the analysis of ways and methods of using deep learning to detect diseases using the example of diabetic retinopathy and analysis of existing disease detection methodologies: single modal model (SM) with input image only and multimodal models with input images and text, such as multimodal model with patient opinion (MP), multimodal model with expert opinion (ME), multimodal model with patient and expert opinion (MPE) and multimodal model with fuzzy expert opinion (MMFE). The influence of additional data, such as the subjective opinion of the “patient” about his state of health and the opinion of the “expert” (which provides “data leakage”), can be useful in some practical situations. The opinions of patients and experts were simulated with additional (augmented) data obtained from simulated questionnaires.

Research was conducted on the metadata augmentation to solve the problem of multiclass classification, which showed that all the created multimodal models (MP, ME, MPE) in comparison with the single modal model (SM) made it possible to achieve various statistically significant improvements in the accuracy of multiclass classification by the value of the area under the curve (AUC) for all classes ranged from 4% to 27%, exceeding the 2-3% standard deviation measured by cross-validation.

Research of the influence of the complexity of the multimodal model on the solution of the multiclass classification task was conducted, which showed that the multimodal model (MP) compared to the single modal model (SM) allowed to achieve various statistically significant improvements in the accuracy of the multiclass classification according to the AUC value for some classes in the range of 15% up to 26% (depending on the complexity of the architecture of the convolutional component of the neural network), which are beyond the standard deviation of 3-8% measured by cross-validation.

Research of fuzzy metadata augmentation for solving the multiclass classification task was conducted, which showed that the multimodal model (MMFE) compared to the

single modal model (SM) achieved various statistically significant improvements in multiclass classification accuracy in terms of AUC ranging from 12% to 26%.

Theoretical estimates of the possible ones are given (the best – when the accuracy by classes increases by the total error of accuracy by individual classes for a single modal model, where the error can be reduced thanks to the application of labeling in an additional modality; intermediate – when the accuracy by classes increases by the total error of accuracy multiplied by the distribution function probabilities, where the possible dependence of the improvement on the average accuracy for a single modal model is demonstrated; and the worst – when the accuracy by class does not improve at all) levels of improvement in the accuracy of multiclass classification using a simple mathematical model with an emphasis on some practical cases.

Based on theoretical evaluations, research of the impact of multimodal metadata addition on the accuracy of multiclass classification using different standard (CIFAR10) and specialized medical (PathMNIST, RetinaMNIST) data sets was conducted, which showed that the accuracy of multimodal models varies significantly depending on the complexity of the data set, sample size and data variability.

A complex method of multimodal analysis of medical data based on deep learning is proposed, which consists in the use of "data leakage" on extreme and similar classes, which allows to increase the accuracy of determining individual classes for solving multiclass classification tasks.

An analysis of the results of using the proposed complex method of multimodal analysis of medical data was conducted, which showed that the proposed method, unlike existing solutions, due to the use of additional modalities based on the metadata augmentation, made it possible to increase the accuracy of determining individual classes for the tasks of multiclass classification of diseases by 4-27% in comparison with the standard single modal approach for the considered identical datasets and neural network architectures.

Keywords: multiclass classification, neural networks, deep learning, metadata augmentation, fuzzy labeling, multimodal model, retina, diabetic retinopathy.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковано основні наукові результати дисертації:

1. **Shulha, M.**, Gordienko, Y., Stirenko, S. (2023). Deep Learning with Metadata Augmentation for Classification of Diabetic Retinopathy Level. In: Shakya, S., Balas, V.E., Haoxiang, W. (eds) Lecture Notes in Networks and Systems, vol 587 (pp. 613-630). Springer, Singapore. ISSN 2367-3389. DOI 10.1007/978-981-19-7874-6_46. Scopus, Q4.
2. **Shulha, M.**, Gordienko, Y., Stirenko, S. (2023). Impact of Multimodal Model Complexity on Classification of Diabetic Retinopathy Level. In: García Márquez, F.P., Jamil, A., Eken, S., Hameed, A.A. (eds) Lecture Notes in Networks and Systems, vol 643 (pp. 168-180). Springer, Cham. ISSN 2367-3389. DOI 10.1007/978-3-031-27099-4_13. Scopus, Q4.
3. Gordienko, Y., **Shulha, M.**, Kochura, Y., Rokovyi, O., Alienin, O., Stirenko, S. (2023). Fuzzy Metadata Augmentation for Multimodal Data Classification. In: Shakya, S., Papakostas, G., Kamel, K.A. (eds) Lecture Notes on Data Engineering and Communications Technologies, vol 166 (pp. 157-172). Springer, Singapore. ISSN 2367-4520. DOI 10.1007/978-981-99-0835-6_11. Scopus, Q3.
4. Gordienko, Y., **Shulha, M.**, Kochura, Y., Rokovyi, O., Alienin, O., Taran, V., Stirenko, S. (2023). Ensemble Knowledge Distillation for Edge Intelligence in Medical Applications. In: Pedrycz, W., Chen, SM. (eds) Studies in Computational Intelligence, vol 1100 (pp. 135-168). Springer, Cham. ISSN 1860-9503. DOI 10.1007/978-3-031-32095-8_5. Scopus, Q4.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	16
ВСТУП.....	19
РОЗДІЛ 1	
ОСОБЛИВОСТІ ВИКОРИСТАННЯ ГЛИБОКОГО НАВЧАННЯ В МЕДИЦИНІ.....	23
1.1. Огляд поточного стану і викликів.....	23
1.2. Соціально-економічні особливості важких захворювань.....	27
1.3. Глибоке навчання в охороні здоров'я.....	28
1.4. Архітектури систем для виявлення зображень за допомогою глибокого навчання.....	30
1.4.1. Системи на основі сканування.....	31
1.4.2. Наскрізні системи.....	33
1.4.3. Навчання детекторам згорткових мереж.....	35
1.5. Виявлення та локалізація в медицині.....	37
1.5.1. Локалізація анатомічного орієнтира.....	37
1.5.2. Виявлення площини зображення.....	38
1.5.3. Виявлення патології.....	39
1.6. Висновки до розділу 1.....	41
РОЗДІЛ 2	
СПОСОБИ ВИКОРИСТАННЯ ГЛИБОКОГО НАВЧАННЯ В МЕДИЦИНІ НА ПРИКЛАДІ ВИЯВЛЕННЯ ДІАБЕТИЧНОЇ РЕТИНОПАТІЇ.....	43
2.1. Автоматизоване виявлення діабетичної ретинопатії.....	43
2.1.1. Типи уражень діабетичної ретинопатії.....	43
2.1.2. Стадії діабетичної ретинопатії.....	45
2.1.3. Класифікація діабетичної ретинопатії.....	45
2.1.4. Завдання виявлення та загальна структура.....	47

2.2. Набори даних зображень очного дна сітківки.....	48
2.3. Методології виявлення діабетичної ретинопатії за допомогою глибокого навчання.....	52
2.3.1. Контрольоване навчання.....	52
2.3.2. Самоконтрольоване навчання.....	55
2.4. Аналіз результатів використання існуючих методологій виявлення діабетичної ретинопатії.....	58
2.4.1. Контрольоване навчання.....	58
2.4.2. Самоконтрольоване навчання.....	59
2.5. Висновки до розділу 2.....	59

РОЗДІЛ 3

МЕТОД МУЛЬТИМОДАЛЬНОГО АНАЛІЗУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ.....	61
------------------------------------------------------------------	----

3.1. Доповнення метаданих для вирішення завдання багатокласової класифікації... 61	
3.1.1. Набір даних.....	62
3.1.2. Моделі.....	66
3.1.3. Робочий процес.....	68
3.1.4. Програмні засоби.....	68
3.2. Вплив складності мультимодальної моделі на вирішення завдання багатокласової класифікації.....	69
3.2.1. Набір даних.....	70
3.2.2. Моделі.....	70
3.2.3. Робочий процес.....	71
3.2.4. Програмні засоби.....	72
3.3. Недетерміноване доповнення метаданих для вирішення завдання багатокласової класифікації.....	72

3.3.1. Набори даних.....	73
3.3.2. Моделі і доповнення недетермінованих метаданих.....	75
3.3.3. Програмні засоби.....	76
3.4. Вплив мультимодального доповнення метаданих на точність багатокласової класифікації.....	76
3.4.1. Набори даних.....	76
3.4.2. Моделі теоретичної оцінки впливу мультимодального доповнення метаданих на точність багатокласової класифікації.....	77
3.4.3. Програмні засоби.....	78
3.4.4. Матриця невідповідностей: Табличний і матричний вигляд.....	79
3.4.5. Матриця невідповідностей: Властивості.....	80
3.4.6. Матриця невідповідностей: Практичні випадки.....	85
3.4.6.1. Рівномірна функція розподілу ймовірностей.....	85
3.4.6.2. Зміщена рівномірна функція розподілу ймовірностей.....	86
3.4.7. Методологія дослідження.....	88
3.5. Висновки до розділу 3.....	89

РОЗДІЛ 4

АНАЛІЗ РЕЗУЛЬТАТІВ МЕТОДУ МУЛЬТИМОДАЛЬНОГО АНАЛІЗУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ.....

90

4.1. Доповнення метаданих для вирішення завдання багатокласової класифікації...	90
4.1.1. Одномодальна модель (SM).....	90
4.1.2. Мультимодальна модель з думкою пацієнта (MP).....	91
4.1.3. Мультимодальна модель з думкою експерта (ME).....	92
4.1.4. Мультимодальна модель з думкою пацієнта та експерта (MPE).....	92
4.1.5. Аналіз отриманих результатів.....	93
4.2. Вплив складності мультимодальної моделі на вирішення завдання	

	15
багатокласової класифікації.....	95
4.2.1. Одномодальна модель (SM).....	95
4.2.2. Мультимодальна модель з думкою пацієнта (MP).....	97
4.2.3. Аналіз отриманих результатів.....	98
4.3. Недетерміноване доповнення метаданих для вирішення завдання багатокласової класифікації.....	102
4.4. Вплив мультимодального доповнення метаданих на точність багатокласової класифікації.....	109
4.5. Висновки до розділу 4.....	113
ВИСНОВКИ.....	115
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	118
ДОДАТОК А	
Код реалізації програмного засобу для проведення дослідження доповнення метаданих для вирішення завдання багатокласової класифікації.....	131
ДОДАТОК Б	
Фрагменти коду реалізації компоненти обробки вхідних даних програмного засобу для проведення дослідження недетермінованого доповнення метаданих для вирішення завдання багатокласової класифікації.....	145
ДОДАТОК В	
Код реалізації програмного засобу для проведення дослідження впливу мультимодального доповнення метаданих на точність багатокласової класифікації.....	149
ДОДАТОК Г	
Схема робочого процесу.....	154
ДОДАТОК Д	
Список публікацій здобувача.....	155

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- 2D (3D) – 2(3)-dimensional
- 3CCD – 3 charge-coupled device
- AlexNet – Alex network
- APTOS – Asia pacific teleophthalmology society
- AUC – Area under curve
- AutoML – Automated machine learning
- CAB – Category attention block
- CABNet – Category attention block network
- CGCN – Convolutional graph classification network
- CIFAR10 (CIFAR100) – Canadian institute for advanced research
- CNN – Convolutional neural network
- CRC-VAL-HE-7K – Colorectal cancer-validation-histology-7K
- CRF – Conditional random field
- CV – Cross-validation
- DDR – Dataset of diabetic retinopathy
- DeepDR – Deep learning-based drug repositioning
- DeepDRiD – Deep diabetic retinopathy image dataset
- DenseNet121 (DenseNet201) – Dense network
- DRiDB – Diabetic retinopathy image database
- EfficientNetB0 (EfficientNetB7, EfficientNetV2L, EfficientNetV2S) – Efficient network
- EyePACS – Eye picture archive communication system
- FMDA – Fuzzy metadata augmentation
- FN – False negative
- FP – False positive
- GAB – General attention block
- GABNet – General attention block network
- GoogLeNet – Google network

GSSL – Generalized self-supervised learning

HAM10000 – Human against machine

HOG – Histogram of oriented gradients

IDRID – Indian diabetic retinopathy image dataset

ImageNet – Image network

InceptionResNetV2 – Inception residual network

IoU – Intersection over union

LBP – Local binary pattern

LSTM – Long short-term memory

LUNA16 – Lung nodule analysis

MCG-Net – Multi-label classification network based on graph convolutional network

MCGS-Net – Multi-label classification network based on graph convolutional network and self-supervised learning.

MedMNIST (RetinaMNIST, BloodMNIST, DermaMNIST, PathMNIST) – Medical (Retina, Blood, Dermatoscope, Pathology) Modified national institute of standards and technology

MESSIDOR – Methods to evaluate segmentation and indexing techniques in the field of retinal ophthalmology

MNIST – Modified national institute of standards and technology

MobileNetV2 – Mobile network

NASNetLarge (NASNetMobile) – Neural architecture search network

NCT-CRC-HE-100K – National center for tumor diseases colorectal cancer-histology-100K

ODIR – Ocular disease recognition

OHEM – Online hard example mining

OoCV – Out of cross-validation

PIN – Patch-based iterative network

QALY – Quality-adjusted life year

R-CNN – Region based convolutional neural network

ReLU – Rectified linear unit

ResNet (ResNet50, Resnet152) – Residual network

RFA-BNET – Round-wise features aggregation on bracket-shaped network

R-FCN – Region based fully convolutional network

RGB – Red, green, blue

ROC – Receiver operating characteristic

RPN – Region proposal network

SIFT – Scale-invariant feature transform

SSD – Single shot detector

SSL – Self-supervised learning

SVM – Support vector machine

TP – True positive

t-SNE – t-distributed stochastic neighbor embedding

U-Net – University [of Freiburg] network

VGG (VGG16, VGG19) – Visual geometry group

WRN – Wide residual network

Xception – Extreme inception

YOLO – You look only once

ДР – Діабетична ретинопатія

КТ – Комп'ютерна томографія

МРТ – Магнітно-резонансна томографія

НПДР – Непроліферативна діабетична ретинопатія

ПДР – Проліферативна діабетична ретинопатія

ПЕТ – Позитронно-емісійна томографія

УЗД – Ультразвукова діагностика

ВСТУП

Актуальність теми. Медична спільнота активно використовує потенціал методів штучного інтелекту в медицині. Протягом останніх років глибокі нейронні мережі продемонстрували свою ефективність і придатність для автоматизованого виявлення та класифікації захворювань. Методи, пов'язані зі штучним інтелектом, можуть автоматизувати та значно пришвидшити програми скринінгу шляхом автоматизованої обробки медичних даних без залучення медичного персоналу на етапі скринінгу. Однією із актуальних проблем є діагностика діабетичної ретинопатії, яка є частим наслідком цукрового діабету та причиною погіршення зору, що вражає доросле населення у всьому світі. Для ранньої діагностики та своєчасного лікування діабетичної ретинопатії з метою запобігання сліпоти необхідний періодичний огляд очей. Цього можна досягти шляхом розробки та впровадження ефективних програм комп'ютерного скринінгу. То ж наразі є актуальним створення підходу з використанням методів комп'ютерного зору та глибоких нейронних мереж для надання сучасних передових медичних послуг на основі штучного інтелекту, спрямованих на класифікацію діабетичної ретинопатії.

Зв'язок роботи з науковими програмами, планами, темами. Тема дисертаційної роботи входить в план наукової роботи затвердженому на кафедрі обчислювальної техніки КПІ ім. Ігоря Сікорського, що враховує розпорядження Кабінету Міністрів України від 2 грудня 2020 р. № 1556-р про схвалення Концепції розвитку штучного інтелекту в Україні. Запропоновані у дисертації методи використані у науково-дослідних проектах:

- “Knowledge At the Tip of Your fingers: Clinical Knowledge for Humanity” (КАТУ), № 101017453, який фінансується в рамках програми Horizon 2020 Європейської Унії;
- “Розробка гібридних моделей штучного інтелекту для аналізу мультимодальних медичних даних”, № К-І-144, який фінансується Міністерством освіти і науки України.

Мета і завдання дослідження. Метою дисертаційної роботи є підвищення

точності багатокласової класифікації захворювань глибокими нейронними мережами, шляхом розробки методу мультимодального аналізу медичних даних.

Об’єкт дослідження – процеси аналізу вимог, розробки, впровадження і супроводження програмного забезпечення для мультимодального аналізу медичних даних на основі глибокого навчання, які сприяють підвищенню точності визначення окремих класів для завдання багатокласової класифікації медичних даних.

Предмет дослідження – методи і моделі розробки і супроводу програмного забезпечення для мультимодального аналізу медичних даних на основі глибоких нейронних мереж різної архітектури і способів організації додаткових модальностей.

Методи дослідження. Методичною основою дослідження є системне опрацювання та аналіз теоретичного матеріалу, присвяченого підвищенню точності вирішення завдання багатокласової класифікації, шляхом аналізу медичних даних за допомогою моделей глибоких нейронних мереж. В процесі даного дослідження були використані методи статистичного аналізу, метрики точності обробки даних, а також програмні засоби для високопродуктивних обчислень.

Для досягнення вказаної мети необхідно вирішити такі завдання:

- дослідити сучасні методи та системи виявлення та багатокласової класифікації захворювань на прикладі діабетичної ретинопатії;
- виявити шляхи покращення існуючих методів та систем виявлення та багатокласової класифікації захворювань на прикладі діабетичної ретинопатії;
- розробити метод мультимодального аналізу медичних даних на основі доповнення метаданих;
- розвинути спосіб підвищення точності багатокласової класифікації завдяки використанню методу мультимодального аналізу для різних архітектур згорткової компоненти нейронної мережі;
- розробити метод недетермінованого штучного доповнення метаданих;
- розвинути спосіб підвищення точності багатокласової класифікації завдяки використанню методу недетермінованого штучного доповнення метаданих для різних стандартних і спеціалізованих медичних наборів даних;

- розробити математичний опис для оцінки підвищення точності багатокласової класифікації на різних стандартних і спеціалізованих медичних наборах даних.

Наукова новизна отриманих результатів. Розроблено метод та програмні засоби мультимодального аналізу медичних даних на основі глибокого навчання для виявлення та багатокласової класифікації захворювань.

Вперше розроблено метод мультимодального аналізу медичних даних на основі доповнення метаданих, який на відміну від існуючих рішень за рахунок використання додаткової модальності дозволяє підвищити точність визначення окремих класів для завдань багатокласової класифікації.

Набув подальшого розвитку спосіб універсального підвищення точності багатокласової класифікації завдяки використанню методу мультимодального аналізу для різних архітектур згорткової компоненти нейронної мережі, що на відміну від існуючих рішень за рахунок використання додаткової модальності дозволяє підвищити точність визначення окремих класів в широкому діапазоні обчислювальних можливостей.

Вперше розроблено метод недетермінованого штучного доповнення метаданих, який на відміну від існуючих рішень за рахунок використання “витоку даних” на крайніх і подібних класах дозволяє підвищити точність визначення окремих інших класів.

Набув подальшого розвитку спосіб універсального підвищення точності багатокласової класифікації завдяки використанню методу недетермінованого штучного доповнення метаданих для різних стандартних і спеціалізованих медичних наборів даних, що на відміну від існуючих рішень за рахунок використання додаткової модальності дозволяє підвищити точність визначення окремих класів медичних зображень.

Вперше запропоновано математичний опис для оцінки підвищення точності багатокласової класифікації на різних стандартних і спеціалізованих медичних наборах даних, що на відміну від існуючих рішень за рахунок формального математичного опису дозволяє кількісно оцінити і якісно інтерпретувати вплив

мультимодального доповнення метаданих на точність визначення окремих класів медичних зображень.

Практичне значення отриманих результатів. Метод мультимодального аналізу медичних даних на основі глибокого навчання дозволяє підвищити точність визначення ступеня захворювання для діагностики діабетичної ретинопатії для медичних працівників, а також збільшити пропускну здатність обробки великих обсягів медичних даних шляхом автоматизації цього процесу для підвищення ефективності праці медичних працівників.

Особистий внесок здобувача. Дисертація є результатом самостійних наукових досліджень, в яких вкладено авторський підхід у розробці методу та програмних засобів мультимодального аналізу медичних даних на основі глибокого навчання. Наукові положення та основні результати, які містяться в дисертації, отримані здобувачем самостійно у процесі науково-дослідницької роботи.

Апробація результатів дисертації. Основні результати роботи опубліковано та обговорено на міжнародних та всеукраїнських наукових конференціях, зокрема на: 3rd International Conference on Sustainable Expert Systems, ICSES 2022 (м. Лалітпур, Непал, 2022); International Conference on Computing, Intelligence and Data Analytics, ICCIDA 2022 (м. Ізміт, Туреччина, 2022); 9th International Conference on Soft Computing & Machine Intelligence, ICSCMI 2022 (м. Торонто, Канада, 2022); International Conference on Trends in Sustainable Computing and Machine Intelligence, ICTSM 2023 (м. Бангкок, Таїланд, 2023).

Публікації. За результатами дисертаційних досліджень опубліковано 4 наукові статі, що входять до наступних наукометричних баз даних з міжнародним індексом цитування: Scopus – 4.

Структура і обсяг роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел із 110 найменувань. Загальний обсяг дисертації становить 156 сторінок, з яких 103 сторінки основного тексту, 5 додатків на 25 сторінках, та містить 33 рисунки, 36 формул, 15 таблиць.

РОЗДІЛ 1

ОСОБЛИВОСТІ ВИКОРИСТАННЯ ГЛИБОКОГО НАВЧАННЯ В МЕДИЦИНІ

1.1. Огляд поточного стану і викликів

Прогнозується, що до 2030 року в усьому світі буде 429 мільйонів хворих на цукровий діабет у результаті зростання рівня ожиріння, збільшення тривалості життя та кращого скринінгу на діабет [1].

Очікувалося, що у 2000 році в Індії 32 мільйони людей хворітимуть на цукровий діабет, а до 2030 року він, ймовірно, вплине на 79 мільйонів [2]. Приблизно у 0,7 мільйона індійців розвинеться проліферативна діабетична ретинопатія, а у 1,8 мільйона – клінічно виражений макулярний набряк, якщо поширеність ускладнень залишиться на такому ж рівні. Досі неясно, чи призведуть зміни способу життя, викликані урбанізацією в Індії та інших країнах, що розвиваються, до неконтрольованої глікемії, артеріального тиску та рівня ліпідів, а також до більш високої частоти важкої діабетичної ретинопатії у людей з діабетом 2 типу. У розвинених країнах покращення надання медичної допомоги зменшило частоту погіршення зору серед населення.

Інтенсивне метаболічне управління уповільнює розвиток і прогресування діабетичної ретинопатії, відповідно до результатів дослідження «Лікування діабету та дослідження ускладнень». Але незважаючи на покращення в лікуванні діабету, проблеми можуть виникати з різних причин. У майже 20% діабетиків, які пройшли інтенсивний метаболічний контроль, може розвиватись більш проліферативна діабетична ретинопатія та інші ускладнення після 30 років [3], і ідеального метаболічного контролю важко досягти через підвищений ризик гіпоглікемії та нефізіологічний шлях введення інсуліну. У дослідженні «Епідеміологія втручань і ускладнень діабету» лише 17% людей з дослідження «Лікування діабету та дослідження ускладнень» мали рівень глікованого гемоглобіну нижче 7% під час останнього візиту.

Ресурси, необхідні для здійснення ефективного контролю діабету, як правило, недоступні в менш розвинених країнах. Тому попередженню проблем слід приділяти більший пріоритет. Для цього необхідно краще розуміти механізми, за допомогою яких діабет ушкоджує сітківку, а також більш ефективні способи ідентифікації ретинопатії.

Згідно з оцінками, у 2000 році у світі був 171 мільйон дорослих, які страждали на цукровий діабет (2,8% населення світу), а до 2030 року це число зросте до 366 мільйонів (4,4% передбачуваного населення світу) в результаті урбанізації, зростання населення, старіння, збільшення ожиріння та гіподинамії [4]. Через 20 років майже всі пацієнти з діабетом 1 типу та понад 60% пацієнтів з діабетом 2 типу отримають діабетичну ретинопатію, оскільки частота захворювання зростає з тривалістю діабету [5].

Захворюваність на діабетичну ретинопатію становила 95,9% у пацієнтів із діабетом 1 типу під час 14-річного спостереження за Вісконсинським епідеміологічним дослідженням діабетичної ретинопатії, і вона була дещо вищою у чоловіків, ніж у жінок (97,7% проти 94%) [6]. У осіб, у яких діагностовано діабет до 30 років, поширеність проліферативної діабетичної ретинопатії становить приблизно 25% після 15 років і 55% після 20 років. У тих, кому діагностовано діабет після 30 років, поширеність проліферативної діабетичної ретинопатії становить 20% після 20 років [7].

Численні дослідження вивчали, наскільки поширені діабетична ретинопатія та діабетичний макулярний набряк у людей з діабетом. Ці дослідження показують широкий діапазон поширення діабетичної ретинопатії від 10% до 41% [8-10]. Однак через відмінності в датах збору даних, віці пацієнтів, тривалості та типі діабету порівняти дослідження складно.

З діапазоном 3–13% поширеність діабетичного макулярного набряку є дещо менш нестабільною. У дослідженні [9] клінічно значущий макулярний набряк був виявлений у 6,2% випадків, а діабетичний макулярний набряк — у 10,4% випадків. Інше дослідження, проведене в Сполучених Штатах, показало, що у людей з діабетом (типу 1 або 2), яким діагноз був поставлений у віці до або після 30 років

відповідно, частота клінічно значущого макулярного набряку становила 5,9% і 7,5%. У Вісконсинському епідеміологічному дослідженні діабетичної ретинопатії частота діабетичного макулярного набряку протягом 14-річного періоду становила 26,1% (29,7% у чоловіків і 22,6% у жінок), тоді як частота клінічно значущого макулярного набряку становила 17% (21,2% у чоловіків і 12,8% у жінок) [6].

Розглядаючи населення в цілому, Національний інститут очей США, використовуючи об'єднані дані 8 популяційних досліджень, підрахував, що 4,1 мільйона дорослих у Сполучених Штатах мають діабетичну ретинопатію (3,4%) і 0,9 мільйона мають загрозову для зору діабетичну ретинопатію (0,8%). Національне опитування про стан здоров'я серед дорослих у Сполучених Штатах, які не перебували в лікарнях, показало, що поширеність діабетичної ретинопатії протягом життя становить 0,7% (1,3 мільйона). За оцінками Decision Resources, Inc., станом на 2016 р. у Сполучених Штатах існує 4,4 мільйона випадків діабетичної ретинопатії; 2,9 млн. в Європі та 1,6 млн. в Японії (табл. 1.1). Лише близько 2/3 цих випадків діагностуються, а 35-45% лікуються, залежно від регіону.

Таблиця 1.1 – Популяція пацієнтів з діабетичною ретинопатією, відсоток діагностованих і пролікованих у США, Європі та Японії, 2006-2016 рр.

Популяція пацієнтів з діабетичною ретинопатією			
	2006	2011	2016
Загальна кількість випадків	7 324 696	8 171 739	8 974 122
Сполучені Штати			
Випадки	3 190 603	3 813 269	4 429 462
Діагностовано (%)	65	70	75
Пройшли лікування (%)	15	40	45
Європа			
Випадки	2 643 552	2 789 058	2 936 861
Діагностовано (%)	60,3	66,9	70,9
Пройшли лікування (%)	10,4	34,2	39,2
Японія			
Випадки	1 490 541	1 569 412	1 607 799
Діагностовано (%)	60	65	70
Пройшли лікування (%)	0	9	35

Незважаючи на те, що захворюваність на цукровий діабет зростає, останні дані показують, що рівень захворюваності на будь-яку нову діабетичну ретинопатію та захворюваність на стадії, що загрожують зору, нижчі, ніж ті, що були зареєстровані 15-20 роками раніше, хоча в цих дослідженнях були відмінності в тривалості діабету. В іспанському дослідженні [11] поширеність діабетичної ретинопатії у пацієнтів з діабетом 2 типу знизилася з 39,4% у 1993 році до 27,5% у 2006 році; хоча зниження не спостерігалось серед пацієнтів з діабетом 1 типу (35,5% у 1993 році та 36,6% у 2006 році).

Наявні прогнози свідчать про те, що кількість людей з діабетичною ретинопатією та небезпечними для зору ускладненнями діабетичної ретинопатії майже потроїться протягом наступних 30 років. Очікується, що кількість людей з діабетичною ретинопатією зросте з 5,5 мільйонів у 2005 році до 16 мільйонів у 2050 році, а кількість людей із загрозовими для зору ускладненнями діабетичної ретинопатії зросте з 1,2 мільйона у 2005 році до 3,4 мільйона у 2050 році. Ці прогнози свідчать про ще більше зростання кількості темношкірих та латиноамериканців з діабетичною ретинопатією та небезпечними для зору ускладненнями діабетичної ретинопатії, особливо серед тих, кому 65 років і старше. За оцінками, у 2050 році 5,9 мільйона білих, 1 мільйон темношкірих і 2,9 мільйона латиноамериканців віком від 65 років матимуть діабетичну ретинопатію. Прогнозований приріст ускладнень діабетичної ретинопатії, що загрожують зору, є значно меншим, але також показує очікуване збільшення до 1 мільйона для білих, 0,4 мільйона для темношкірих і 0,5 мільйона для латиноамериканців віком від 65 років до 2050 року.

Прогнози захворювань очей, пов'язаних з діабетом, базуються на стабільних показниках поширення, отриманих на основі інформації з опублікованих досліджень. У результаті очікується, що зміни в кількості людей з діабетом супроводжуватимуться коливаннями в кількості діабетиків із захворюваннями очей. Природно, що прогнозоване зростання поширення діабету має призвести до збільшення тягаря діабетичного макулярного набряку [12].

1.2. Соціально-економічні особливості важких захворювань

Діабет вражає 5% населення світу, і його поширеність подвоюється з кожним поколінням. За оцінками Міжнародної діабетичної федерації, у 2005 році приблизно 333 мільйони людей у світі у віці від 20 до 79 років мали діабет. У Сполученому Королівстві з населенням понад 60 мільйонів більше ніж 2 мільйонам людей було діагностовано діабет, а приблизно 750 000 людей мають недіагностований діабет. У цій популяції понад 250 000 хворих на захворювання 1 типу, тоді як 1,8 мільйона мають діабет 2 типу, причому остання група становить близько 90% пацієнтів з діабетом [13].

З населенням 10 144 940 чоловік Португалія має 1 059 642 діагностованих випадків діабету [14] та приблизно 105 000 випадків ретинопатії [15], або близько 10% населення. За оцінками, у 3,5% усіх діагностованих пацієнтів з діабетом може розвинутися клінічно значущий діабетичний макулярний набряк, один із наслідків діабетичної ретинопатії, який може погіршити зір.

Люди, які хочуть знизити свої шанси на діабетичні ускладнення, включаючи діабетичну ретинопатію та загрозу зору, повинні мати доступ до ефективної медичної допомоги. Метою медикаментозної терапії є зниження артеріального тиску та рівня цукру, а також раннє виявлення та лікування проблем.

Цукровий діабет має високу ціну як для людини, так і для держави. Для фізичних осіб існують як витрати на охорону здоров'я, так і витрати на фармацевтику, причому останні включають витрати на придбання медичних послуг і втрату заробітку. За оцінками, у 2007 році витрати на лікування цукрового діабету та його ускладнень у всьому світі склали 232 мільярди доларів, причому понад 50% цих витрат було витрачено на Сполучені Штати та 25% на Європу.

Сучасні стратегії управління зосереджені на зниженні факторів ризику та скринінгу. Згідно з рекомендаціями Міжнародної діабетичної федерації, кожна людина з цукровим діабетом повинна щорічно проходити пряму офтальмоскопію або знімок очного дна, а також професійний огляд очей для перевірки гостроти зору.

Згідно з даними за 2008 рік, частковий зір і сліпота завдають загальних

економічних збитків Сполученому Королівству 22 мільярди фунтів стерлінгів, а прямі медичні витрати становлять 2,14 мільярда фунтів стерлінгів. Згідно з дослідженням, 3,5% дорослого населення Сполученого Королівства мали діабетичну ретинопатію, що становить загалом 1,8 мільйона осіб із частковим зором і сліпотою. Потенційно уникнути або знизити ці витрати можна за допомогою раннього виявлення та лікування діабетичної ретинопатії.

У Сполучених Штатах скринінг і лікування очних захворювань у пацієнтів з цукровим діабетом коштує \$3190 за рік здорового якісного життя (QALY). Ця вартість є середньозваженим (на основі поширеності пацієнтів з діабетом різних підтипів) економічної ефективності виявлення та лікування діабетичного захворювання очей у пацієнтів з інсулінозалежним цукровим діабетом (\$1996 за QALY), пацієнтів з інсуліннезалежним діабетом, які використовують інсулін для контролю глікемії (\$2933 за QALY) і тих, хто страждає на неінсуліновий цукровий діабет, які не використовують інсулін для контролю глікемії (\$3530 за QALY) [16].

Діабетична ретинопатія є особливо гарним прикладом захворювання, яке потребує розгляду в перспективі, що включає планування медичної допомоги [17].

1.3. Глибоке навчання в охороні здоров'я

Для широкого спектру робіт у галузі медицини точне та швидке виявлення анатомічних або хворих структур або маркерів має вирішальне значення. Наприклад, визначення місцезнаходження анатомічних орієнтирів має важливе значення для реєстрації зображення або для налаштування об'ємної сегментації органів. Важливим першим етапом у створенні систем автоматизованого виявлення та діагностики, популярність яких зросла за останні кілька десятиліть, є виявлення уражень. Крім того, використання алгоритмів виявлення може допомогти зі структурним моніторингом під час лікування або з локалізацією відповідних площин зображення по всьому об'єму даних медичного зображення.

Методи виявлення об'єктів відрізняються від алгоритмів класифікації тим, що вони виводять обмежувальні прямокутники елементів або структур, щоб

визначити не лише їх присутність на зображенні, але й їх точне розташування в ньому. Подібні завдання включають локалізацію та виявлення; однак, хоча алгоритми виявлення об'єктів можуть визначити наявність і розташування кількох об'єктів, присутніх на зображенні (рис. 1.1), алгоритми локалізації зазвичай ідентифікують один об'єкт на зображенні. У результаті алгоритм виявлення створюватиме обмежувальні прямокутники для кожного об'єкта на зображенні разом із інформацією про тип об'єкта, який містить кожне поле (зазвичай значення ймовірності, що вказує, чи належить об'єкт до певного класу).

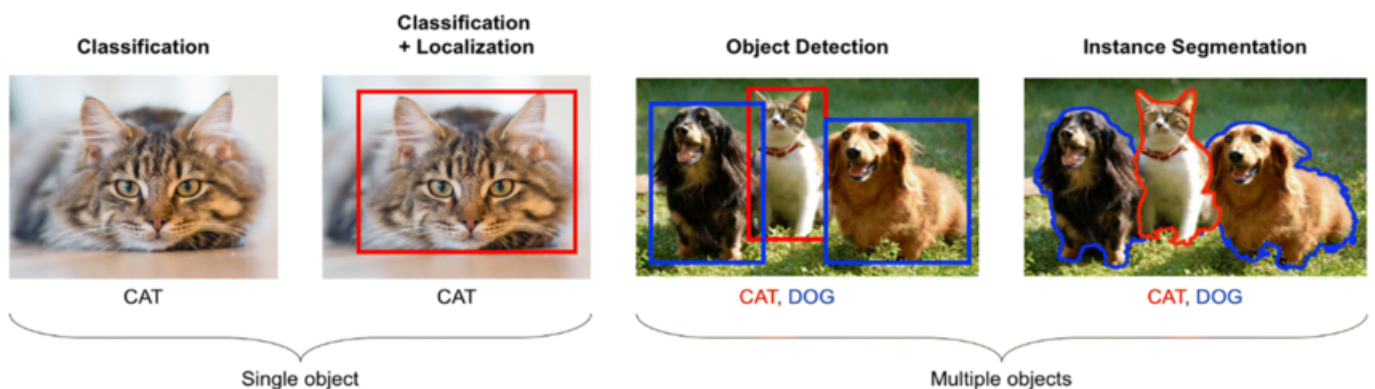


Рисунок 1.1 – Різниця між класифікацією, локалізацією, виявленням і сегментацією

У комп'ютерному баченні алгоритми виявлення об'єктів зазвичай використовують такі два кроки:

- пропозиція регіону, яка є процесом видалення різних патчів із зображення в пошуках потенційних патчів, які можуть містити предмет інтересу. За допомогою техніки ковзного вікна або спеціалізованих алгоритмів пропозиції регіону все зображення можна відсканувати та сегментувати на фрагменти, щоб визначити області, які, швидше за все, містять певні об'єкти.
- класифікація зібраних патчів для створення обмежувальних рамок, які мають певну ймовірність містити об'єкт.

Цей тип алгоритмів відомий як системи на основі сканування. Традиційно алгоритми виділення ознак (наприклад, SIFT [18], HOG [19], LBP [20], вейвлети Хаара [21] або перетворення Хафа [22]) використовуються для характеристики фрагментів зображення, і ці ознаки передаються в класифікатори, такі як метод опорних векторів (SVM) або випадкові ліси. Цей підхід використовувався в кількох задачах виявлення об'єктів у медичній сфері [23–25].

Ефективність згорткових нейронних мереж (CNN) у класифікації зображень [26] надихнула глибоке навчання для розпізнавання зображень, яке спирається на ознаки, отримані CNN, а не на набори створених вручну ознак. CNN на основі регіону (R-CNN) була запропонована у 2014 році [27]. Для пропозиції регіону вони використовували популярний алгоритм, відомий як вибіркового пошуку [28]. Отримані фрагменти зображень потім передаються в CNN, AlexNet [26], для виділення ознак, і, нарешті, SVM використовується для класифікації. З того часу було розроблено та оцінено багато інших моделей виявлення зображень на основі CNN. Сучасні методи глибокого навчання для виявлення об'єктів усувають крок пропозиції регіону або отримують пропозиції регіону безпосередньо з карти функцій замість зображення, підвищуючи швидкість і перевершуючи результати традиційних алгоритмів виявлення об'єктів.

Однак, порівняно зі сферою комп'ютерного бачення, існують значні проблеми, пов'язані з предметною областю, які завдання виявлення в медичній візуалізації має вирішити, наприклад, нестача значних баз даних із анотованими даними. Це вимагає, щоб вчені, які працюють у медичній промисловості, модифікували або створювали алгоритми виявлення, спеціально адаптовані до цієї дисципліни.

Детектори об'єктів на основі глибокого навчання використовувалися в широкому діапазоні патологій, наприклад, раку молочної залози, раку передміхурової залози та ретинопатії, а також для локалізації орієнтирів і анатомічних структур, які можна використовувати як орієнтир для реєстрації або сегментації зображення. Далі буде надано огляд різних стратегій локалізації та виявлення об'єктів і орієнтирів на медичних зображеннях.

1.4. Архітектури систем для виявлення зображень за допомогою глибокого навчання

Системи на основі сканування є найпоширенішим підходом до виявлення об'єктів. Вони складаються з фази пропозиції регіону, за якою слідує етап

класифікації, що вирішує завдання виявлення як проблему класифікації за фрагментами. Перші засновані на CNN підходи для виявлення об'єктів були засновані на цій схемі, вводячи CNN або для виділення ознак, або для класифікації патчів (рис. 1.2). У деяких випадках CNN використовується як для пропозиції регіону, так і для класифікації, тоді як інші автори використовують різні методи комп'ютерного бачення для створення пропозицій регіону, а CNN використовуються лише для етапу класифікації. У цих системах кожен модуль має бути навчений окремо (для створення регіону та для класифікації).

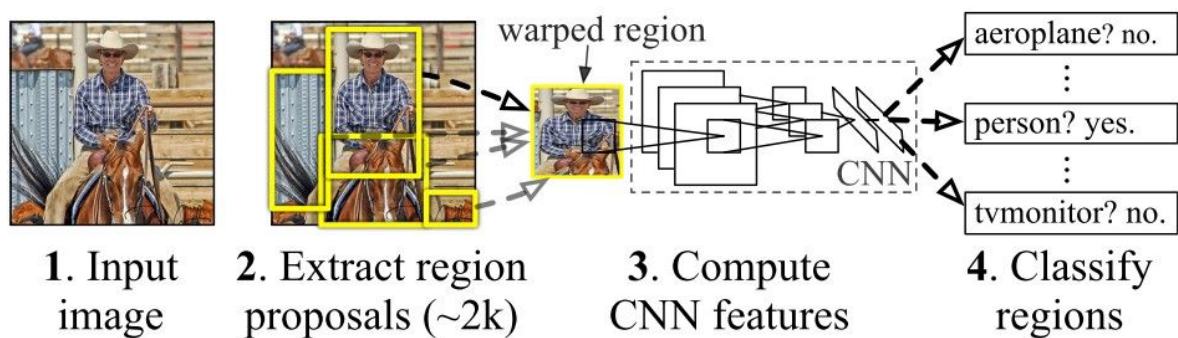


Рисунок 1.2 – Огляд R-CNN системи

З іншого боку, останні роботи з виявлення об'єктів пропонують наскрізні системи з прямим відображенням між вхідним зображенням і вихідними прогнозами. Ці архітектури можна оптимізувати наскрізно та використовувати функції всього зображення для прогнозування обмежувальних рамок замість розгляду лише незалежних патчів. Щоб передбачити обмежувальні прямокутники, CNN мають не лише виконати завдання класифікації, а й регресію координат обмежувальної рамки. Зазвичай це включається в багатозадачну втрату, яка поєднує втрати класифікації та втрати регресії.

1.4.1. Системи на основі сканування

Ці системи покладаються на крок пропозиції регіону для вилучення патчів зображення, а потім класифікують ці патчі відповідно до об'єкта, який вони містять. CNN можна використовувати як для виділення регіону, так і для класифікації (як у Faster R-CNN), або лише для одного з задіяних кроків. Мотивований успіхом мережі

AlexNet [26] для завдань класифікації, Гхіршік та ін. запропонували систему виявлення об'єктів R-CNN [27]. Вони зберігають традиційну схему пропонування регіону, виділення ознак і класифікації та вводять лише CNN для етапу виділення ознак. Вони використовували популярний алгоритм пропозиції регіонів, вибіркового пошуку [28], щоб створити різні регіони інтересу, і передали ці патчі в CNN для вилучення характеристик. Функції, обчислені мережею, потім використовуються для класифікації фрагментів зображення за допомогою лінійної SVM і застосовують жадібний немаксимальний алгоритм придушення для вибору остаточних обмежувальних рамок.

Хоча R-CNN отримала хороші результати, її основним недоліком є час обчислення. Вона виконує пряму передачу ConvNet для кожної пропозиції об'єкта без спільного використання обчислень. Таким чином, у [29] Fast R-CNN було розроблено, щоб зменшити це обчислювальне навантаження. Fast R-CNN потрібна лише одна головна CNN для обробки всього зображення, але вона все ще покладається на алгоритм вибіркового пошуку для створення пропозицій регіону. Ці пропозиції регіону потім подаються в мережу, яка в цьому випадку приймає все зображення разом з патчами як вхідні дані та безпосередньо виводить оцінки ймовірності.

Faster R-CNN [30] включає мережу регіональних пропозицій (RPN) для безпосереднього створення регіональних пропозицій і прогнозування обмежувальних рамок без необхідності використання вибіркового пошуку чи інших алгоритмів. RPN приймає зображення як вхідні дані та виводить набір прямокутних об'єктів із оцінкою об'єктності. Це робиться шляхом ковзання невеликої мережі по згортковій карті ознак і генерації нижньовимірного вектора, який подається на два повністю зв'язані шари: шар коробкової регресії та шар коробкової класифікації. Пропозиції параметризовані відносно k опорних блоків, так що вони розташовані в центрі ковзного вікна та пов'язані з масштабом і співвідношенням сторін. У кожному розташуванні ковзного вікна k пропозицій області прогнозуються одночасно, тому рівень регресії кодує координати k коробок, а рівень класифікації виводить $2k$ балів, ймовірність об'єкта чи не об'єкта для кожної пропозиції

(рис. 1.3). Отримана модель являє собою комбінацію Fast R-CNN і RPN, де RPN навчаються наскрізного обміну згортковими функціями з Fast R-CNN, таким чином зменшуючи витрати на обчислення під час тестування. Схема навчання має чергуватися між тонким налаштуванням для завдання пропозиції регіону та тонким налаштуванням для виявлення об'єктів, зберігаючи пропозиції фіксованими.

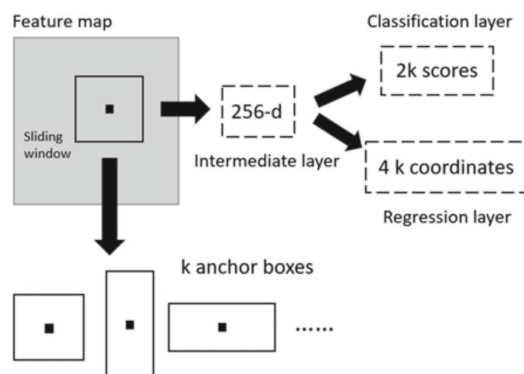


Рисунок 1.3 – Генерація пропозицій регіонів у Faster R-CNN

У медичній сфері наявність першого та окремого кроку пропозиції регіону дозволяє використовувати алгоритми пропозиції регіону, спеціально розроблені для конкретного завдання або модальності зображення, а потім навчити CNN лише для етапу класифікації. Наприклад, Саварді та ін. [31] використовували алгоритм пропозиції регіону, який використовував їхні знання про фізичні ефекти, які гемоліз виробляє на плівці крові, викликаючи варіації світла, щоб виділити ділянки, які, швидше за все, відповідають гемолітичним областям. Сетіо та ін. [32] використовували детектори-кандидати, спеціально розроблені для твердих, субсолідних і великих легеневи́х вузлів, а Терамото та ін. [33] виявили легеневі вузлики, поєднуючи ПЕТ і КТ за допомогою алгоритмів пропозиції регіону, специфічних для кожного типу зображення.

1.4.2. Наскрізнi системи

Наскрізне навчання стосується навчання можливо складної системи навчання шляхом застосування градієнтного навчання до системи в цілому. Ці підходи використовують єдину модель, щоб забезпечити повне зворотне розповсюдження для навчання та логічного висновку, що призводить до систем, які навчаються

наскрізно та безпосередньо відображають вхідне зображення на вихід. На відміну від методів на основі сканування, наскрізні системи не покладаються на попередню пропозицію об'єкта.

Редмон та ін. [34] переформулював виявлення об'єктів як єдину задачу регресії, безпосередньо передбачаючи обмежувальні рамки та ймовірності класів із зображень. Вони об'єднують компоненти виявлення об'єктів в єдину нейронну мережу під назвою YOLO. Вхідне зображення розділене на сітку, і кожна комірка сітки передбачає обмежувальні прямокутники та оцінки достовірності для цих рамок (рис. 1.4). Передбачається велика кількість обмежувальних рамок, тому метод немаксимального придушення має бути застосований наприкінці мережі, щоб об'єднати сильно перекриваючі обмежувальні рамки одного об'єкта. Ця система є швидкою та дозволяє виявляти в реальному часі.

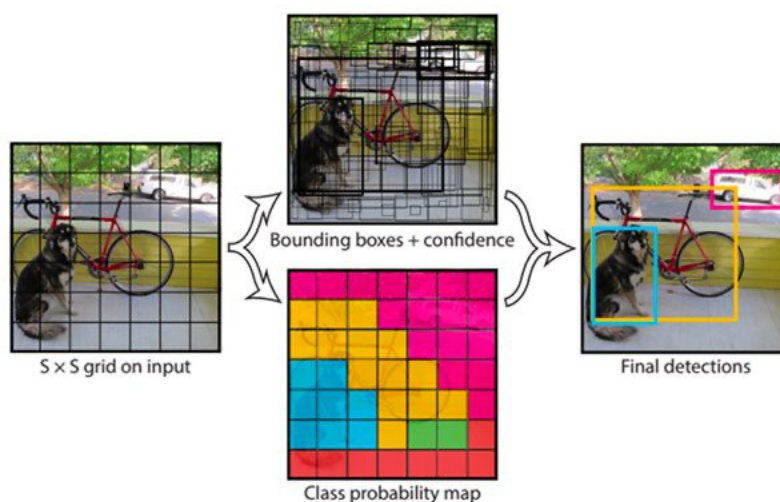


Рисунок 1.4 – Модель YOLO

Що стосується виявлення орієнтирів, архітектури кодера-декодера, які популярні для сегментації зображення, також можна використовувати з невеликими модифікаціями. У системах кодер-декодер мережа кодувальників відображає вхідне зображення на представлення ознак, а мережа декодувальників приймає це представлення ознак як вхідні дані, виробляє вихідні дані та відображає їх назад у пікселях. Щоб адаптувати цей підхід до виявлення та локалізації орієнтирів, локалізація орієнтирів розглядається як задача попиксельної регресії теплової карти, де теплові карти для навчання створюються шляхом застосування функцій Гауса в істинних положеннях ключових точок. Популярна мережа для сегментації медичних

зображень, U-Net [35], широко застосовувалася з невеликими модифікаціями для локалізації орієнтирів у медичних зображеннях [36–38], а також інші архітектури кодера-декодера [39].

Створення анотованих наборів даних є дорогим і особливо складним у сфері медицини, оскільки зазвичай потрібні експертні знання. Це спонукало деяких авторів спробувати підходи зі слабким контролем [40–42] для виявлення об'єктів. Їм потрібні лише мітки на рівні зображення, і вони знаходять розташування орієнтирів або об'єктів за допомогою карт уваги, створених мережею. Коли мережа навчиться розрізняти зображення з різними мітками, розпізнавальні моделі уражень вивчаються автоматично, і ці функції можна використовувати для оцінки розташування уражень.

1.4.3. Навчання детекторам згорткових мереж

Вибір відповідної функції втрат, яку потрібно оптимізувати, і вибір найкращої стратегії навчання є дуже важливими факторами для конвергенції мережі та можуть мати великий вплив на продуктивність. Деякі з раніше описаних підходів вимагають складної стратегії навчання, наприклад Faster R-CNN [30] має чергувати точне налаштування для RPN і точне налаштування для виявлення об'єктів, тоді як інші мережі можна навчати безпосередньо наскрізно. Показники помилок або функції втрат, які потрібно оптимізувати, також відрізняються для кожного підходу, але загалом вони мають бути розроблені для кількісного визначення як помилки класифікації, так і помилки локалізації. Перетин над об'єднанням (IoU) – це метрика, яка зазвичай використовується для оцінки результатів виявлення об'єктів. Це значення між 0 і 1, яке представляє область перекриття між передбачуваним прямокутником і базовим полем істинності (рис. 1.5).

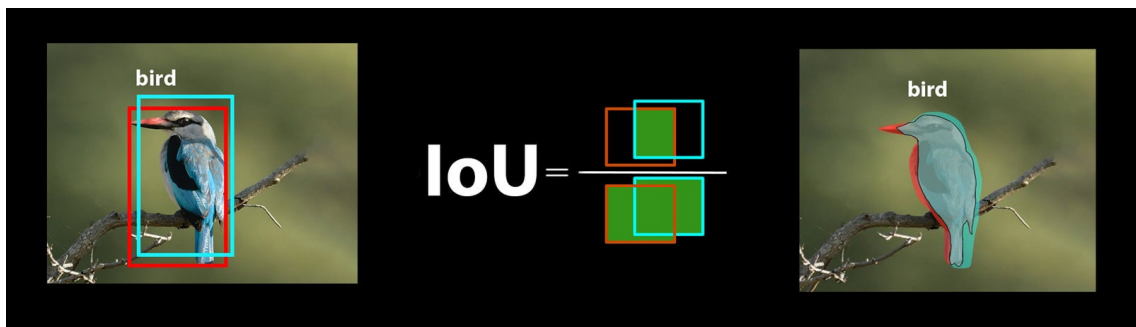


Рисунок 1.5 – Приклад обчислення IoU

Мережі виявлення мають не лише передбачити обмежувальну рамку, але й класифікувати об'єкт, що міститься в цій рамці. Тому пропонуються багатозадачні втрати, які поєднують метрики для класифікації з метриками для локалізації, враховуючи IoU.

Для RPN [30] втрати класифікації є логарифмічними втратами для двох класів (об'єкт чи ні), а втрати регресії є гладкими L1 над параметризованими координатами обмежувальної рамки. Вони активуються лише для позитивних пропозицій регіонів, тобто регіонів із найвищим IoU з полем правдивості землі або з IoU вище 0,7. У YOLO [34] оптимізували помилку суми квадратів, помилку локалізації зважування та помилку класифікації. Кожна комірка сітки передбачає кілька обмежувальних прямокутників (рис. 1.4), обмежувальна рамка вважається відповідальною за виявлення об'єкта наземного істинного об'єкта, якщо вона має найвищий IoU з усіх предикторів у цій комірці сітки.

Інше важливе міркування щодо навчання детекторів об'єктів стосується роботи з сильним дисбалансом класів. Коли генеруються патчі або пропозиції щодо регіонів, кількість запропонованих регіонів завжди значно перевищує кількість об'єктів, отже, негативних збігів більше, ніж позитивних. Стратегії жорсткої негативної вибірки були включені до схеми навчання CNN для вирішення цієї проблеми. R-CNN [27] включає традиційну жорстку негативну вибірку для SVM [43], де модель навчається з початковою підмножиною негативних зразків, а потім негативні приклади, неправильно класифіковані початковою моделлю, використовуються для формування нового набору негативних прикладів. Ефективним способом жорсткого негативного видобутку є використання регіональних детекторів CNN – жорсткий онлайн-видобуток прикладів (OHEM) [44].

Він полягає в обчисленні втрат для всіх пропозицій регіонів і виборі тих, які мають найвищий бал; лише ця невелика кількість регіонів інтересу вибирається для оновлення моделі. Це робиться для кожної міні-партії (кожної ітерації стохастичного градієнтного спуску). Схему навчання OHEM можна використовувати з різними архітектурами, як Faster R-CNN, R-FCN і SSD.

Нарешті, стратегії збільшення даних також використовуються для виявлення, щоб вирішити проблему відсутності великих баз даних зображень для навчання. Різні перетворення, такі як масштабування інтенсивності, пружні деформації, обертання або трансляції [32, 36, 38, 45] застосовуються до зображень, щоб створити нові зразки для надання їх мережі.

1.5. Виявлення та локалізація в медицині

В контексті систем на основі глибокого навчання класифікація зображень передбачає призначення мітки класу зображенню, тоді як локалізація об'єктів передбачає малювання обмежувальної рамки навколо одного або кількох об'єктів на зображенні. Виявлення об'єктів є більш складним і поєднує в собі ці два завдання та малює обмежувальну рамку навколо кожного цільового об'єкта та призначає їм мітку класу. Разом усі ці проблеми називаються розпізнаванням об'єктів.

1.5.1. Локалізація анатомічного орієнтира

Хоча більшість літератури зосереджено на виявленні патологій, виявлення анатомічних орієнтирів також важливе для багатьох завдань аналізу медичних зображень, таких як реєстрація на основі орієнтирів, ініціалізація алгоритмів сегментації зображення та виділення клінічно значущих площин із 3D-об'ємів.

Виявлення орієнтирів зазвичай виконується за наскрізною схемою з використанням добре відомих архітектур сегментації типу кодер-декодер. Паєр та ін. [36] використовували структуру CNN (4 різні архітектури порівнювали, включаючи U-Net і нещодавно запропоновану Spatial-Configuration Net), щоб

отримати анатомічні орієнтири з рентгенівських знімків рук і МРТ рук. Вони безпосередньо навчили CNN наскрізним способом регресії теплових карт для орієнтирів. На рентгенівських знімках виявлено 37 анатомічних орієнтирів, на МРТ – 28. Мадер та ін. [37] застосували подібний підхід, використовуючи U-Net та CRF. Вони позначили 16 точок для кожного ребра на рентгенівських знімках грудної клітини. U-Net використовувався для створення гіпотез локалізації, які пізніше були вдосконалені за допомогою CRF для оцінки просторової інформації. Меєр та ін. [38] також використовували архітектуру кодера-декодера, засновану на U-Net, для регресії відстані від кожного місця зображення до цікавих орієнтирів, оптичного диска сітківки та ямки. Таким чином вони змогли спільно виявити обидві структури.

Інший підхід до виявлення орієнтирів полягає у застосуванні методів на основі патчів. Кай та ін. [46] об'єднали функції зображення з різних модальностей, а саме МРТ та КТ, щоб покращити розпізнавання та локалізацію хребців. Вони використовували CNN, щоб об'єднати різні модальності та витягти особливості з патчів зображень і передати їх у класифікатор SVM. Лі та ін. [47] запропонували ітеративну мережу на основі патчів (PIN) для виявлення 10 анатомічних орієнтирів на УЗД голови плода. Чжен та ін. [48] дотримувалися двоетапного підходу з неглибокою мережею для створення регіонів-кандидатів і CNN для класифікації в контексті локалізації біфуркації сонної артерії на КТ ший.

1.5.2. Виявлення площини зображення

Виявлення певної площини зображення в межах цілого медичного обсягу даних є важливим завданням, яке може заощадити клініцистам тривалий час пошуку, і для якого запропоновано декілька рішень. Чен та ін. використовували CNN для локалізації стандартної площини в УЗД плода [49]. Вони використовували перенесене навчання, щоб зменшити надмірну підгонку, спричинену невеликою кількістю навчальних зразків. Баумгартнер та ін. [50] також запропонували систему для виявлення стандартної площини сканування плоду. Вони використовували VGG16 [51] як основну архітектуру для розробки SonNet, мережі, яка може виявляти

13 стандартних видів плоду та забезпечувати локалізацію структур плода через обмежувальну рамку, використовуючи слабкий контроль, заснований лише на мітках на рівні зображення [50]. Кумар та ін. вирішили ту саму проблему, використовуючи карти помітності та CNN [52].

1.5.3. Виявлення патології

Медичні зображення зазвичай отримують для діагностичних процедур, тому виявлення наявності патології є дуже важливим завданням аналізу медичних зображень. Локалізація та класифікація ракових уражень, яка зазвичай є складною, оскільки доброякісні та злоякісні пухлини можуть мати схожий вигляд, є одним із ключових застосувань виявлення об'єктів у медичній сфері.

Створення випробування для виявлення легеневих вузликів на зображеннях КТ, LUNA16 [53], полегшило роботу в цій галузі. Виявлення цих вузликів має важливе значення для діагностики раку легень, однак це може бути складно через високу варіабельність форми, розміру та текстури. Сетіо та ін. [32] запропонували двовимірний підхід, який складається з порогового значення щільності з подальшим морфологічним розкриттям, щоб отримати кандидатів, вилучити набір 2D патчів з різноорієнтованих площин для кожного кандидата та подати їх у двовимірну згортку. Дінг та ін. [54] використовували 2D Faster R-CNN [30] і наступний 3D CNN для хибно позитивного зменшення. Доу та ін. [55] використовували повністю згорнуту мережу 3D і прийняли стратегію онлайн-фільтрації зразків, щоб збільшити частку важких навчальних зразків, щоб підвищити точність і впоратися з дисбалансом між складними та легкими зразками. Чжу та ін. [56] використовували Faster R-CNN, як Дінг та ін. [54], але їхній підхід є повністю тривимірним, із 3D-швидшим R-CNN для генерації вузликів-кандидатів і подібною до U-Net архітектурою 3D-кодера-декодера для вивчення функцій. Терамото та ін. [33] включили інформацію з ПЕТ, крім КТ, ідентифікуючи кандидатів окремо на ПЕТ і КТ, а потім поєднавши регіони-кандидати, отримані з двох зображень.

Комбінація декількох методів візуалізації або послідовностей актуальна для

виявлення багатьох патологій. Результати мультипараметричної МРТ показали високу кореляцію з гістопатологічними дослідженнями раку передміхурової залози, а інформація, отримана за допомогою різних послідовностей МРТ, може бути вирішальною для оцінки злоякісності виявленого ураження. Кіралі та ін. [39] використовували багатоканальні кодери-декодери зображення в зображення з ядрами Гауса, розташованими в ключових точках, і різними вихідними каналами для представлення різних класів пухлин. Янг та ін. [45] використовували мережу, навчену слабо контрольованому способу, зменшуючи вартість генерації анотацій. Вони модифікували GoogLeNet для створення карт реагування на рак, моделювання кількох класів і об'єднання мультимодальної інформації з аналого-цифрового перетворювача і T2w.

CNN також застосовувалися для виявлення раку молочної залози. Платанія та ін. [57] адаптували модель YOLO для мамографічних зображень. Подібним чином Аль-Масні та ін. [58] запропонували систему автоматизованої діагностики, також засновану на YOLO, для одночасного виявлення маси грудей і класифікації в цифровій мамографії. Коої та ін. [59] також працювали з мамографіями, але замість цього вони використовували зменшену версію моделі VGG. Лі та ін. [47] працювали з гістологічними зображеннями для діагностики раку молочної залози. Вони використовували модель на основі Faster R-CNN для виявлення мітозу, включаючи модель глибокої перевірки на основі ResNet [60] для підвищення точності.

Приклад використання мереж виявлення для подальшої ініціалізації сегментації можна знайти в [61]. Вони використовували мережу виявлення, щоб знайти область інтересу, яка містить тромб на зображеннях КТ-ангіографії, щоб пізніше виконати сегментацію в межах виділеної області.

Інші клінічні застосування включають виявлення уражень головного мозку за допомогою МРТ [40, 62], бета-гемолізу за гістологічними зображеннями [31], виявлення уражень при розсіяному склерозі за допомогою МРТ [63], зон інфаркту міокарда [64] та внутрішньочерепних крововиливів на КТ головного мозку [65].

1.6. Висновки до розділу 1

Однією із актуальних проблем є діагностика захворювань, що вражають доросле населення у всьому світі. Індустрія охорони здоров'я активно використовує технології штучного інтелекту. Глибокі нейронні мережі нещодавно довели свою ефективність і придатність для автоматизованого виявлення та діагностики. Завдяки автономній обробці медичних даних без допомоги медичного персоналу на етапі скринінгу підходи, пов'язані зі штучним інтелектом, можуть автоматизувати та значно прискорити програми скринінгу.

Виявлення об'єктів є важливим завданням обробки для багатьох медичних програм, особливо для виявлення уражень. Глибоке навчання дозволяє автоматизовано локалізувати підозрілі утворення за допомогою кількох методів візуалізації, таких як КТ, МРТ або УЗД, а іноді навіть класифікувати ураження як доброякісні чи злоякісні, допомагаючи спеціалістам і надаючи цінний внесок у комп'ютерні системи виявлення. Іншим актуальним застосуванням систем виявлення на основі згорткових нейронних мереж є автоматизована локалізація площини інтересу, яка може заощадити багато часу при спробі знайти значні структури в межах цілих об'ємів. Крім того, локалізація та виявлення анатомічних орієнтирів можуть допомогти ініціалізувати інші алгоритми обробки зображень, такі як сегментація.

Існують різні підходи до виявлення об'єктів, які можна застосувати до обробки медичних зображень. Системи на основі сканування покладаються на етап пропозиції регіону для створення патчів, які пізніше класифікуються відповідно до об'єкта, який вони містять, тоді як новіші системи безпосередньо генерують обмежувальні рамки з усього вхідного зображення, підвищуючи точність і дозволяючи виявлення в реальному часі. Однак навчання згорткових нейронних мереж для виявлення медичних зображень все ще стикається з важливими проблемами. Основним обмеженням є відсутність великих загальнодоступних баз даних, які можна використовувати для навчання або передачі навчання. Крім того, коли метою є виявлення патологічних структур, виникає проблема класового

дисбалансу, оскільки зазвичай є більше даних від здорових пацієнтів, ніж від конкретної патології. Для вирішення цієї проблеми зазвичай використовуються стратегії збільшення даних, а також стратегії видобутку важких прикладів. Деякі автори спробували підходи зі слабким наглядом, зменшивши вартість створення анотованих баз даних. Необхідно докласти більше зусиль для створення доступних баз даних і розробки стратегій навчання, які дозволяють використовувати слабо анотовані дані, анотації з шумом та неконтрольоване навчання.

Діабет є серйозною проблемою здоров'я, яка вражає все більше людей у всьому світі внаслідок прогресуючого старіння населення та поширення ожиріння внаслідок збільшення доступу до солодких продуктів у всьому світі.

Діабетична ретинопатія є найчастішим і найстрашнішим ускладненням цукрового діабету, а отже, має величезний вплив на суспільство. Це стає ще більш актуальним, якщо усвідомити, що діабетична ретинопатія, яка нині є найпоширенішою причиною сліпоти серед дорослих працездатного віку в західному світі, поступово охопить увесь світ у міру покращення економічних умов у бідніших частинах світу.

Діабетичний макулярний набряк є найчастішим ускладненням діабетичної ретинопатії та найчастішою причиною втрати зору внаслідок діабету. Повідомлялося, що протягом 14-річного періоду частота клінічно значущого набряку макули становить 17%. Ідентифікація очей/пацієнтів із ризиком розвитку клінічно значущого макулярного набряку та втрати зору, а також розуміння його причин і розвитку є основоположним для відповідного лікування та, нарешті, для уникнення втрати зору через діабет.

Діабетична ретинопатія вже визнана основною причиною сліпоти, але глобальні тенденції в галузі охорони здоров'я та економіки вказують на її зростаючу актуальність у найближчому майбутньому.

Отже, діабетична ретинопатія є серйозною проблемою охорони здоров'я, для вирішення якої необхідно створити точну і ефективну систему раннього виявлення захворювання на основі машинного навчання.

РОЗДІЛ 2

СПОСОБИ ВИКОРИСТАННЯ ГЛИБОКОГО НАВЧАННЯ В МЕДИЦИНІ НА ПРИКЛАДІ ВИЯВЛЕННЯ ДІАБЕТИЧНОЇ РЕТИНОПАТІЇ

2.1. Автоматизоване виявлення діабетичної ретинопатії

Скринінг сітківки сприяє ранньому виявленню діабетичної ретинопатії та своєчасному лікуванню. Щоб полегшити процес скринінгу, слід мати систему глибокого навчання, яка зможе виявляти ранні та пізні стадії діабетичної ретинопатії.

2.1.1. Типи уражень діабетичної ретинопатії

Найбільш ранніми клінічними ознаками діабетичної ретинопатії та ураження сітківки є мікроаневризми, які є розширенням мікроциркуляторного русла, що утворюється внаслідок руйнування внутрішньої еластичної пластинки. Мікроаневризми сітківки погіршують зір через локальну втрату функції ендотеліального бар'єру, викликаючи витік і набряк сітківки. Мікроаневризми невеликі (зазвичай менше 125 мікрон у діаметрі) і виглядають як червоні плями з різкими краями. Коли стінки слабких капілярів розриваються, кровотеча викликає крововиливи, які схожі на мікроаневризми, але більші і мають нерівний край, вони мають різний вигляд залежно від того, в який шар сітківки вони просочуються. Осколкові крововиливи виникають у поверхневих шарах сітківки і викликають поверхневу кровотечу у формі полум'я. Тоді як точкові і крапкові крововиливи відбуваються в більш глибоких шарах сітківки. Більше витоку з пошкоджених капілярів може спричинити ексудат, який зазвичай виглядає жовтим і має неправильну форму в сітківці. Розрізняють два види ексудату: твердий і м'який. Тверді ексудати – це ліпопротеїни та інші білки, що виходять з аномальних судин сітківки (білі або біло-жовті з різкими краями). Вони часто організовані у вигляді блоків або круглих кілець і розташовані у зовнішньому шарі сітківки. З іншого боку, м'який ексудат або ватоподібні плями – це невеликі білувато-сірі хмароподібні

форми, які виникають при оклюзії артеріоли. За яскравістю ексудат відрізняється від мікроаневризми і крововиливів. Мікроаневризми та крововиливи є темними вогнищами, а ексудати – яскравими. Варіації діаметра вен сітківки називаються венозним утворенням. Це зазвичай відбувається на пізніх стадіях непроліферативної діабетичної ретинопатії. Через неможливість використання глюкози звичайними шляхами активуються альтернативні шляхи кровотворення, що викликає синтез таких елементів, як сорбіт, і сприяє розвитку змін у мікроциркуляторному руслі. Внутрішньоретинальні мікросудинні аномалії є прикладом, вони представляють або розширення вже існуючих капілярів, або фактичне зростання нових кровоносних судин у сітківці. Коли судини сітківки виділяються і ростуть у напрямку до склоподібного тіла, це називається неоваскуляризацією. Макулярний набряк виникає, коли капіляри сітківки стають проникними і навколо макули відбувається витік. Це може призвести до потовщення сітківки або твердих ексудатів, що розвиваються або в межах одного діаметра диска від центру макули (фовеа), або залучаючи фовеа, яка відповідає за центральний зір [66].

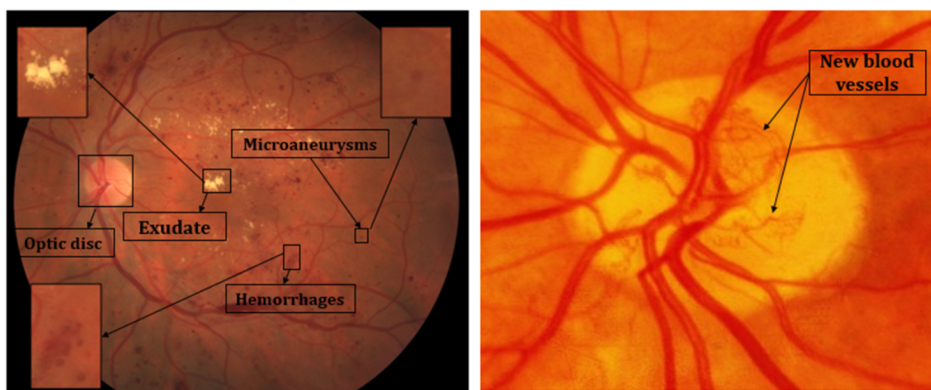


Рисунок 2.1 – Диск зорового нерва та патологічні зміни очного дна, спричинені діабетичною ретинопатією

Важливим об'єктом, який відіграє суттєву роль у виявленні діабетичної ретинопатії, є диск зорового нерва, який характеризується найбільшим контрастом між ділянками округлої форми. Диск зорового нерва використовується як орієнтир і система відліку для діагностики серйозних патологій очей, таких як глаукома, ямка диска зорового нерва, друзи диска зорового нерва, а також для перевірки будь-якої неоваскуляризації на диску. Диск зорового нерва також використовується для

визначення інших структур, таких як ямка. У нормальній сітківці краї диска зорового нерва чіткі та чітко визначені (рис. 2.1).

2.1.2. Стадії діабетичної ретинопатії

За ступенем тяжкості діабетичну ретинопатію можна класифікувати на два основні класи: непроліферативну (НПДР) та проліферативну (ПДР) [67, 68]. Непроліферативна діабетична ретинопатія – це рання стадія, під час якої діабет починає пошкоджувати дрібні кровоносні судини сітківки; це дуже часто зустрічається у людей з діабетом. Ці судини починають викидати рідину і кров, викликаючи набряк сітківки. З плином часу припухлість або набряк потовщують сітківку, викликаючи розмитість зору. Клінічним ознакою цієї стадії є принаймні одна мікроаневризма або крововилив з твердим ексудатом або без нього. Проліферативна діабетична ретинопатія є прогресуючою стадією, яка призводить до росту нових кровоносних судин; як така, вона характеризується аномальною судинною проліферацією в сітківці в напрямку до склоподібного тіла. Ці крихкі нові кровоносні судини можуть кровоточити в порожнину склоподібного тіла та спричиняти серйозну втрату зору через крововилив у склоподібне тіло. Вони також можуть додатково спричинити тракцію сітківки, оскільки вони зазвичай ростуть із фіброзно-судинною мережею навколо них, що може призвести до тракційного відшарування сітківки [66].

2.1.3. Класифікація діабетичної ретинопатії

Обстеження та скринінг сітківки за допомогою офтальмоскопії зазвичай вимагає розширених зіниць, кваліфікованого екзаменатора та візиту до офтальмолога, наприклад оптометриста, для оцінки та класифікації патології. Оцінка є життєво важливою діяльністю в програмі скринінгу діабетичної ретинопатії для діагностики захворювань сітківки. Це інтенсивна процедура, яка потребує навченої робочої сили та комп'ютерних екранів відповідного розміру.

Таблиця 2.1 – Шотландський протокол класифікації діабетичної ретинопатії

Клас	Особливості
R0: Відсутність ДР	Відсутність аномалій
R1: Легка НПДР	Тільки мікроаневризми
R2: Помірна НПДР	Більше, ніж просто мікроаневризми, але менше, ніж важка НПДР
R3: Важка НПДР	<ul style="list-style-type: none"> • Більше 20 крововиливів у кожному квадранті • Венозні кульки в двох квадрантах • Інtrarетинальні мікросудинні аномалії
R4: ПДР	<ul style="list-style-type: none"> • Будь-які нові судини на диску зорового нерва або в іншому місці • Крововиливи в склоподібне тіло/преретинальні крововиливи

Такі оцінювачі, як оптометристи або добре навчені техніки, виконують важливе завдання для лікування захворювань очей, які потенційно призводять до сліпоти, такі як вікова дегенерація жовтої плями та діабетичні захворювання очей. Зазвичай отримують немідріатичні зображення очного дна, але якщо зображення нечітке через будь-яке непрозоре середовище, то мідріатичні краплі використовують для розширення зіниці, намагаючись покращити якість зображення. Усі оцінювачі повинні пройти спеціальну підготовку на основі протоколу скринінгу, щоб гарантувати, що зображення очного дна оцінюються стандартизованим способом. Вони повинні витратити час на навчання, щоб виявляти та підтверджувати випадки наявності чи відсутності аномалії патології, а також диференціювати рівні виявленої патології та приймати рішення про направлення або повернення для повторного виклику на основі узгодженого інтервалу. Існують різні системи для класифікації судинних змін при діабетичній ретинопатії, такі як класифікація Американської академії офтальмології, яка була введена в дослідженні раннього лікування діабетичної ретинопатії [69] і шотландський протокол класифікації діабетичної ретинопатії (табл. 2.1, рис. 2.2), де лише одне поле береться на око, яке зосереджено на ямці [70].

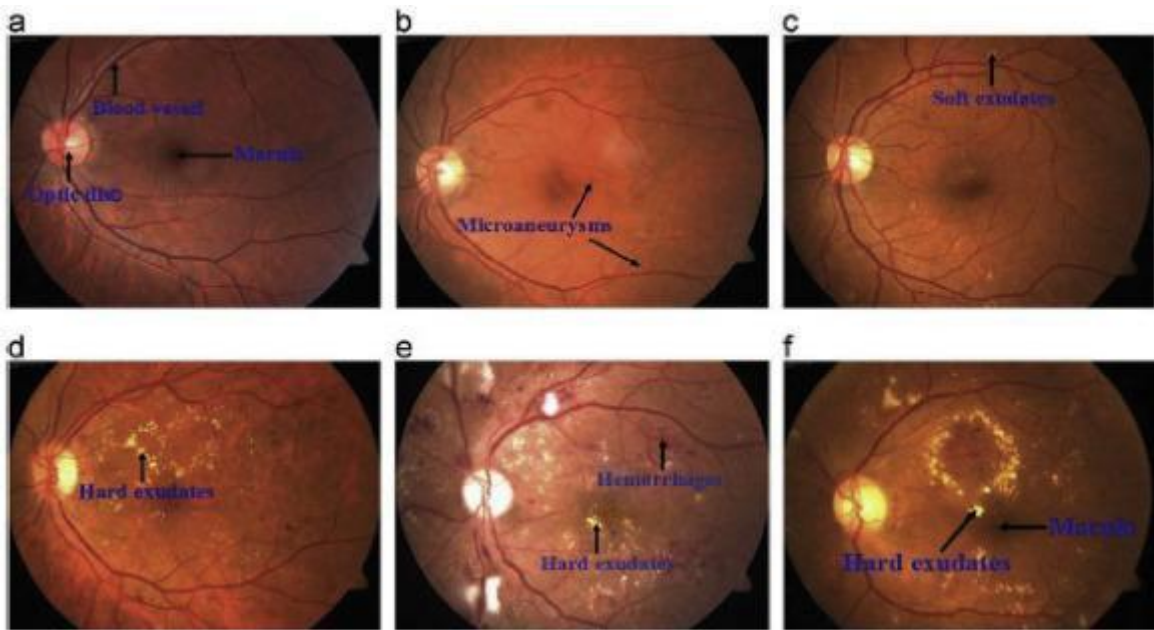


Рис. 2.2 – Стадії ДР: (а) відсутність ДР, (б) легка НПДР, (с) помірна НПДР, (д) важка НПДР, (е) ПДР, (ф) макулярний набряк

Автоматизовані методи виявлення діабетичної ретинопатії економлять кошти та час і є більш ефективними, ніж ручна діагностика. Ручна діагностика схильна до помилкових діагнозів і вимагає більше зусиль, ніж автоматизовані методи.

2.1.4. Завдання виявлення та загальна структура

На високому рівні виявлення діабетичної ретинопатії класифікується на два завдання: виявлення на основі рівня ураження та виявлення на основі зображення. При виявленні на основі рівня ураження виявляється кожне ураження та визначається їх локалізація, оскільки кількість уражень та їх розташування мають вирішальне значення для оцінки рівня тяжкості діабетичної ретинопатії. З іншого боку, виявлення на основі зображення фокусується на оцінці на основі зображення та є більш цікавим з точки зору скринінгу, оскільки воно оцінює лише наявність ознак діабетичної ретинопатії [71]. Виявлення на основі рівня ураження зазвичай включає дві фази: (і) виявлення ураження та/або сегментація та (іі) класифікація ураження. Спочатку на знімках очного дна виявляються такі ураження, як мікроаневризми, крововиливи, тверді та м'які ексудати, і точна область ураження локалізується. Це складне завдання, оскільки зображення очного дна сітківки

містять інші об'єкти подібного вигляду, наприклад червоні точки та кровоносні судини. Для цього завдання зазвичай потрібен глобальний і локальний контекст для точної локалізації та сегментації. Фаза виявлення дає потенційні регіони інтересу, але вони також включають помилкові спрацьовування. Фаза класифікації уражень використовується для видалення помилкових позитивних результатів. Виявлення на основі зображення – це завдання скринінгу зображення, яке класифікує дане зображення очного дна як нормальне або має ознаки діабетичної ретинопатії. Це одна з перших областей медичної діагностики, в яку глибинне навчання зробило значний внесок [72].

Загальна основа для виявлення, сегментації та класифікації включає конкретні етапи попередньої обробки, виділення/відбору ознак, вибір відповідного методу класифікації та, нарешті, оцінку результатів. Системи класифікації діабетичної ретинопатії можна розділити на два типи відповідно до процедури навчання: контрольоване та неконтрольоване навчання. У контрольованому навчанні система навчається з використанням позначених даних для висновку про функціональне відображення [73, 74]. З іншого боку, методи неконтрольованого навчання мають тенденцію самостійно виявляти приховані шаблони за властивостями непозначених прикладів відповідно до їх подібності [75]. На відміну від підходів, розроблених вручну на основі функцій, підходи до глибокого навчання об'єднують усі етапи в єдину структуру й автоматично вивчають функції та навчають систему наскрізним способом.

2.2. Набори даних зображень очного дна сітківки

Було створено значна кількість наборів даних, що складаються із зображень очного дна сітківки, щоб навчити та перевірити алгоритми для різних завдань виявлення діабетичної ретинопатії. Нижче наведено огляд загальнодоступних контрольних наборів даних, які згадуються в даній роботі.

2.2.1. MESSIDOR

Набір даних був розроблений у рамках дослідницької програми MESSIDOR, що фінансується Міністерством досліджень і оборони Франції [76]. Він був зібраний трьома офтальмологічними відділеннями за допомогою кольорової відеокамери 3CCD, встановленої на немідріатичному ретинографі Topcon TRC NW6 з полем зору 45°. Експерти-офтальмологи надали два типи анотацій на рівні зображення: ступені діабетичної ретинопатії та рівні ризику макулярного набряку. Ступені діабетичної ретинопатії:

- 0: жодного ризику: ($\# \text{мікроаневризми} = 0$) І ($\# \text{крововилив} = 0$);
- 1: ($0 < \# \text{мікроаневризми} < 5$) І ($\# \text{крововилив} = 0$);
- 2: ($5 < \# \text{мікроаневризми} < 15$) АБО ($0 < \# \text{крововилив} < 5$) І ($\text{неоваскуляризація} = 0$);
- 3: ($\# \text{мікроаневризми} < 15$) АБО ($\# \text{крововилив} < 5$) АБО ($\text{неоваскуляризація} = 1$).

Рівні ризику макулярного набряку такі:

- 0: жодного ризику;
- 1: найкоротша відстань між макулою та твердим ексудатом $>$ одного діаметра сосочка;
- 2: найкоротша відстань між макулою та твердим ексудатом $<$ одного діаметра сосочка.

2.2.2. E-ophtha

Набір даних був представлений проектом E-ophtha, що фінансується французьким дослідницьким агентством [77]. Він надає розташування мікроаневризм і ексудатів, які виявили два офтальмологи. Перший офтальмолог окреслив місця, які перевірів і оглянув другий офтальмолог. База даних складається з двох наборів даних: E-ophtha ексудат та E-ophtha мікроаневризма. Набір E-ophtha ексудату містить 47 зображень із 12 278 ексудатами та 35 здоровими зображеннями.

Кілька зображень здорових контрольних прикладів містять структури, які можуть легко ввести в оману методи виявлення ексудату, такі як відображення та оптичні артефакти. З іншого боку, E-ophtha мікроаневризму містить 148 зображень із 1306 мікроаневризмами та 233 здоровими зображеннями.

2.2.3. Kaggle

Набір даних складається з великого набору зображень сітківки ока з високою роздільною здатністю, зроблених у різних умовах і наданих клініками EyePACS [78]. Анотація рівня зображення була надана експертами-офтальмологами, і кожному зображенню було присвоєно ступінь діабетичної ретинопатії за шкалою від 0 до 4 таким чином: жодного ризику (0) – легкий (1) – помірний (2) – важкий (3) – ПДР (4).

2.2.4. DRiDB

База даних зображень діабетичної ретинопатії [83] була отримана в університетській лікарні в Загребі та створена для подолання недоліків у попередніх наборах даних, таких як класифікація та обмежена кількість спостерігачів. Зображення були зроблені з полем зору 45° та відібрані експертами, і показано, що симптоми діабетичної ретинопатії варіюються від майже нормальних до випадків, коли видно нові крихкі судини. У цьому наборі даних кожне зображення було оцінено п'ятьма незалежними експертами, щоб відзначити виявлення діабетичної ретинопатії. Ці експерти анотували пікселі знахідок і відповідні області мікроаневризму, крововиливів, твердих і м'яких ексудатів, кровоносних судин, диска зорового нерва та макули.

2.2.5. EyePACS-1

Архів зображень ока та система зв'язку [88] – це гнучка протокольна та веб-телемедицина система для скринінгу діабетичної ретинопатії та співпраці між

клініцистами. Зображення очного дна пацієнтів можна легко завантажити на EyePACS сайт. Протокол оцінює наявність і тяжкість дискретних уражень сітківки, пов'язаних з діабетичною ретинопатією. Протокол використовує немідріатичні камери Canon CR-DGi та Canon CR-1, доступ до яких доступний на веб-сайті EyePACS. Ураження класифікуються як мікроаневризми, крововиливи з мікроаневризмами або без них, ватні плями, інtrarетинальні мікросудинні аномалії, венозні кульки, нові судини (нові судини на диску та нові судини в інших місцях), фіброзна проліферація, крововиливи у склоподібне тіло або преретинальні крововиливи та твердий ексудат. Крім того, наявність або відсутність лазерних рубців. Оцінювачі оцінюють кожен тип ураження окремо на кожному зображенні за допомогою онлайн-шаблону оцінки, який фіксує вибір для кожного типу ураження серед “ні” (відсутній), “так” (присутній) або неможливо оцінити.

2.2.6. DeepDR

Великий набір даних зображень сітківки, DeepDR, був запропонований науковому співтовариству в рамках конкурсу IEEE DeepDR Diabetic Retinopathy Image Dataset (DeepDRiD) [94] з 5 класами: жодного ризику (0) – легка НПДР (1) – помірна НПДР (2) – важка НПДР (3) – ПДР (4).

Набір даних був спрямований на подальше підвищення точності та надійності ранньої діагностики на практиці на основі подвійних зображень очного дна з тих самих очей, наприклад, диск зорового нерва як центр і фовеа як центр, щоб класифікувати ураження діабетичної ретинопатії. Очікувані результати повинні перевершити сучасні моделі, створені за допомогою однопоекції зображень очного дна. Крім того, зображення очного дна різної якості були включені в набір даних DeepDR, щоб відобразити реальний сценарій на практиці. Ця ініціатива була підтримана багатьма іншими подібними змаганнями, спрямованими на сприяння креативності та популярності підходів, пов'язаних із глибоким навчанням, у сфері охорони здоров'я та частково для класифікації діабетичної ретинопатії.

2.2.7. MedMNIST v2

Це масштабна колекція стандартизованих біомедичних зображень, подібна до MNIST, включаючи 12 наборів даних для двовимірних зображень і 6 наборів даних для тривимірних зображень, нещодавно була запропонована для дослідницьких і освітніх цілей [95, 96]. Зображення мають відповідні класифікаційні мітки, тому користувачам не потрібні додаткові знання. Набір даних розроблено для виконання класифікації легких дво- і тривимірних зображень з різними масштабами набору даних (від 10^2 до 10^5) і різними завданнями (бінарна/багатокласова класифікація, порядкова регресія та визначення міток).

2.3. Методології виявлення діабетичної ретинопатії за допомогою глибокого навчання

Класифікацію діабетичної ретинопатії можна розділити на бінарну класифікацію, яка спрямована на виявлення наявності або відсутності діабетичної ретинопатії, і багатокласову класифікацію, яка визначає точну стадію діабетичної ретинопатії. В даній роботі досліджується багатокласова класифікація.

2.3.1. Контрольоване навчання

Контрольоване навчання є підкатегорією машинного навчання та штучного інтелекту. Воно визначається використанням позначених наборів даних для навчання алгоритмів, які класифікують дані або точно прогнозують результати. Коли вхідні дані надходять у модель, вона коригує свої ваги, доки модель не буде підігнано належним чином, що відбувається як частина процесу перехресної перевірки. Контрольоване навчання допомагає організаціям вирішувати різноманітні проблеми реального світу.

Абрамофф та ін. [107] представили метод виявлення діабетичної ретинопатії за допомогою моделі згорткової нейронної мережі. Під час попередньої обробки

зображення були нормалізовані, а потім мали ширину 299 пікселів для діаметра перед подачею зображень у модель. Модель передбачала навчання 10 згорткових нейронних мереж на основі попередньо навченої архітектури InceptionV3. Класифікація включала 5 класів, а саме: діабетичний макулярний набряк, помірна або гірша діабетична ретинопатія, важка або гірша діабетична ретинопатія або повністю градуйована.

Система, запропонована Жанг та ін. [108] була використана для виявлення діабетичної ретинопатії на їх наборі даних. Їхній набір даних був розділений на чотири класи із загальною кількістю 13 767 зображень. Для попередньої обробки зображень використовували кадрування, зміну розміру, вирівнювання гістограми та адаптивне вирівнювання гістограми. Збільшення зображення було здійснено за допомогою збільшення з подальшим покращенням контрасту за допомогою алгоритму розтягування контрасту, який використовується для темних зображень. Попередньо навчені архітектури згорткової нейронної мережі ResNet50, InceptionV3, InceptionResNetV2, Xception і DenseNets були точно налаштовані для класифікації діабетичної ретинопатії. Нові повністю пов'язані шари були навчені на основі вищезгаданих згорткових нейронних мереж. Потім попередньо навчені шари згорткової нейронної мережі були налаштовані для повторного навчання з наступною інтеграцією моделі.

Модель, що використовує R-FCN з модифікаціями, була запропонована Ванг та ін. [109] для виявлення стадій діабетичної ретинопатії для набору даних MESSIDOR [105], а також їхнього приватного набору даних. Їхні модифікації передбачали модифікацію R-FCN шляхом додавання п'яти мереж пропозицій регіону та мережі піраміди функцій. Збільшення було виконано на навчальних зображеннях із надмірним збільшенням, особливо для зображень приватного набору даних.

Пао та ін. [110] використовували двоканальні нейронні мережі для виділення компонентів очного дна за каналом з подальшим покращенням деталей за допомогою класичного інструменту підвищення різкості під назвою нерізке маскування. Для цієї реалізації було використано набір даних про діабетичну

ретинопатію Kaggle [97] з 21 123 зображеннями очного дна у колірній моделі RGB. Розмір зображень було змінено до (100, 100, 3). Потім вони використали дзеркальне відображення та обертання, щоб отримати загалом 33 000 зображень для експериментів. Щоб скласти 30 000 зображень очного дна для навчального набору, випадковим чином вибирають 15 000 зразків із першої групи 0 класу та ще 15 000 із другої групи, що містить зображення очного дна 1–4 ступеня. Таким же чином 3 000 зображень вибирають для тестового набору. Двоканальна згорткова нейронна мережа, яка використовується для вивчення ознак діабетичної ретинопатії навчається за допомогою зеленого компонента зображення очного дна та ентропійних зображень рівня сірого, які спочатку попередньо обробляються нерізким маскуванням для покращеного виявлення діабетичної ретинопатії. Техніка нерізкого маскування використовується для посилення сірих високочастотних частин і зеленого компонента зображення сітківки. Для кожного каналу використовуються чотири згорткових шари з ядрами 5 на 5 із розмірами карти функцій/кількістю фільтрів 32, 64 і 128 для кожного наступного згорткового шару відповідно. Для кожного шару використовується максимальне об'єднання, функція активації випрямленої лінійної одиниці (ReLU) і шари відсіву, з відпаданням, встановленим на 0,3. Після цього виконується зведення для двох каналів, а зв'язок повнозв'язних шарів використовується для визначення класифікації діабетичної ретинопатії.

Тимченко та ін. [79] використовували багатозадачний підхід до навчання для класифікації класів діабетичної ретинопатії за допомогою архітектури глибокої згорткової нейронної мережі з невеликим декодером. Набір даних Kaggle EyePACS [97] використовувався для попереднього навчання згорткової нейронної мережі. Іншими наборами даних, які були об'єднані для навчального набору, були набір даних IDRID [104], що містить 413 фотографій очного дна, і набір даних MESSIDOR [105], який містить 1200 зображень очного дна. Збільшення кількості зображень включає оптичне викривлення, викривлення сітки, кусково-афінне перетворення, горизонтальне відображення, вертикальне відображення, випадкове обертання, випадковий зсув, випадкове масштабування, зсув значень колірної моделі RGB, випадкову яскравість і контрастність, додатковий гаусівський шум, розмиття,

різкість, тиснення, випадкову гамма та виріз. Модель використовує попередньо навчені згорткові нейронні мережі ImageNet для ініціалізації кодера. Вони використовують три декодери, у яких кожен декодер навчений вирішувати власне завдання на основі витягнутих ознак за допомогою магістралі згорткової нейронної мережі з головками класифікації, регресії та порядкової регресії. У той час як вихідні дані класифікаційної головки є однократно закодованим вектором, де значення 1 означає існування кожного відповідного етапу. Вихід головки регресії – це дійсне число в діапазоні від 0 до 4,5, округлене для позначення різних стадій захворювання. Що стосується головки порядкової регресії, передбачається, що всі точки даних у категорії потрапляють у всі категорії, таким чином прогножуючи всі категорії до цільової категорії. Використання ансамблю з трьох головок і підгонка моделі лінійної регресії до результатів трьох головок дає загальний прогноз. Цей ансамбль базується на послідовному характері захворювання, яке було оцінено за набором даних Kaggle APTOS 2019 [80].

2.3.2. Самоконтрольоване навчання

Методи контрольованого навчання не є найкращим вибором для вирішення будь-якої проблеми, особливо коли дані мають шум. Методи самоконтрольованого навчання є чудовою альтернативою методам контрольованого навчання і можуть використовуватися як доповнення до цих методів. Методи самоконтрольованого навчання менш схильні до індуктивного упередження та можуть використовуватися для обробки міждоменого введення. Проблема методів самоконтрольованого навчання полягає в тому, що вони потребують багато даних, щоб бути ефективними. Це проблема, тому що чим більше у вас даних, тим більше часу потрібно для навчання моделі.

Хі та ін. [85] представили новий блок уваги категорій (CAB), щоб експериментувати з функціями на основі регіонів для кожного відповідного ступеня діабетичної ретинопатії. Ця мережа зазвичай використовується для багатокласової класифікації діабетичної ретинопатії, щоб пом'якшити дисбаланс ступеня

діабетичної ретинопатії в розподілі в більшості загальнодоступних наборів даних, таких як MESSIDOR, EyePACS і DDR. Увага до категорії використовується для доповнення уваги до простору та каналу, щоб дозволити CAB бути вбудованим у різні блоки, не орієнтовані на категорію, для покращення багатокласової класифікації, зокрема класифікації діабетичної ретинопатії для цього застосування. Модель поєднує в собі вищезгаданий CAB з GABNet, натхнений Yu та ін. [86] тобто GAB і CAB, де CABNet пропонується для класифікації діабетичної ретинопатії. GAB може вивчати глобальні ексцентричні особливості класу, ігноруючи такі особливості, як колір і текстура. У поєднанні CABNet фіксує детальні характеристики невеликих уражень, щоб вирішити проблему незбалансованого розподілу даних. Модуль CABNet складається з чотирьох частин, а саме магістралі, глобального блоку уваги (GAB), блоку уваги категорії (CAB) і класифікатора. Модуль уваги складається з GAB і CAB, для яких навчання CABNet проходить наскрізно.

Вхідні зображення очного дна надходять до CABNet, для якого магістральна мережа використовується лише для отримання та вилучення карт функцій у глобальному масштабі. Модель є гнучкою, тому будь-яку архітектуру згорткової нейронної мережі можна використовувати як основу, для якої функції можна витягти з останнього згорткового шару з дуже багатими семантичними характеристиками вхідних зображень очного дна. Карта функцій, отримана з магістралі на попередньому кроці, потім спочатку подається в згортковий шар 1 на 1 для зменшення вхідного каналу, який потім передається як вхідні дані для GAB, а вихід просторової уваги подається як вхідні дані для CAB і, нарешті, до класифікатора для класифікації діабетичної ретинопатії. Для навчання базова модель CABNet має магістральну мережу, попередньо навчену на наборі даних ImageNet. Застосовувані перетворення даних включають випадкові горизонтальні перевороти, вертикальні перевороти та випадкове обертання з вхідними зображеннями розміру (512, 512, 3). Швидкість навчання спочатку була встановлена на рівні 0,005 і систематично зменшувалася з використанням коефіцієнта 0,8 на основі втрати перевірки. Навчання виконується протягом 70 епох за допомогою оптимізатора Адама та функції втрат

крос-ентропії. Було навчено різні базові моделі, і в якості базової моделі використовується найкраща модель з мінімальними втратами перевірки. Розмір пакету було встановлено на 16.

Лін та ін. [87] представили модуль під назвою MCG-Net, який базується на графовій згортковій мережі для ефективного виділення ознак зображень уражень очного дна, які використовуються для багатокласової класифікації, і покращили класифікацію уражень. Щоб покращити узагальнення, модуль покращення представленої MCGS-Net побудовано на основі самоконтрольованого навчання, у якому графова згорткова мережа використовується замість повністю зв'язного шару для кращого захоплення кореляції зображень очного дна як класифікатора. Використання самоконтрольованого навчання призводить до покращення здатності узагальнення згорткової нейронної мережі. Модель має три основні компоненти, які є магістральним модулем для спільного вилучення ознак, модулем CGCN і модулем GSSL. Набори даних Self-Supervision і ODIR передаються в згорткову нейронну мережу для вилучення зображення. Після шару глобального максимального об'єднання отримується вектор ознак. Згодом GSSL дозволяє MCGS-Net навчатися на більшій кількості неанотованих даних, використовуючи техніку самоконтролю, для якої GSSL використовує повністю підключений рівень як класифікатор. Потім CGCN використовує класифікатор із графової згорткової мережі для отримання категорійної кореляції між зображеннями очного дна. Модель використовує загалом три набори даних для конвеєра, а саме набори даних ODIR, SSL і GTest. Для тестування та навчання багатокласової мережі використовувалася набір даних ODIR. Набір даних SSL використовувався для навчання MCGS-Net у самоконтрольований спосіб, у якому анотовані людиною мітки було видалено. Потім набір даних GTest використовувався для тестування здатності до узагальнення в шести різних мережах в абляції, головним чином MCGS-Net і MCG-Net. Дві мережі продемонстрували кращу точність порівняно з ResNet50, DenseNet121, EfficientNetB0.

2.4. Аналіз результатів використання існуючих методології виявлення діабетичної ретинопатії

Для виявлення та багатокласової класифікації захворювання показники AUC, F1 і Каппа використовуються як основний показник для визначення достовірності результатів, тоді як інші показники, такі як точність і запам'ятовування, можна вважати допоміжними тому, що розподіл даних є незбалансованим у кожному наборі даних.

2.4.1. Контрольоване навчання

Дослідження Тимченко та ін. [79], використовує передавальне навчання в архітектурі згорткової нейронної мережі з 3 головками ансамблю (класифікаційна, порядкова і регресійна) і досягає найкращих результатів, використовуючи ансамбль із 20 різних моделей і скороченого середнього 200 п'ятикласових прогнозів для кожного зображення очного дна. Використовуючи збільшення часу навчання, модель досягла 99,3% точності. Якість моделі оцінювали за допомогою скринінгу бінарної класифікації та досягли оцінки F1 99,3%.

Одна з найскладніших моделей, запропонована Жанг та ін. [108] використовує сукупність попередньо навчених мереж, які зрештою перетворюються на цілісну структуру для виявлення діабетичної ретинопатії, але порівняно з Лі та ін. [90] дослідження, яке використовує один шар ResNet50, результати не такі вже й далекі від того, що може бути досягнуто високоякісною мережею ансамблю. Лі та ін. [90] вдалося досягти чутливості, AUC і точності 92%, 96,3% і 92,6% відповідно для набору даних MESSIDOR [105], тоді як досягнута точність для IDRID становила близько 65,1%, про прецизійність або F1 не було повідомлено. Намагаючись зменшити кількість параметрів, які можна навчити, Заго та ін. [102] запропонував мережу VGG16, яка досягає чутливості 94% і AUC 91,2% для набору даних MESSIDOR, на відміну від цього, ця модель може виявити лише діабетичну ретинопатію без будь-яких ознак тяжкості.

2.4.2. Самоконтрольоване навчання

Модель CABNet від Xi та ін. [85], тренуючись із 70 епохами на наборі EYEPACS, змогла узагальнити для MESSIDOR і досягти дуже конкурентоспроможної точності та Каппа 84% і 85,5% відповідно. Спеціальний блок уваги CABNet полегшує інтерпретацію того, як працює модель і які області зображень очного дна сприяють впровадженню. Така архітектура дає CABNet перевагу з точки зору налаштування блоків уваги та додає їй гнучкості.

2.5. Висновки до розділу 2

Хоча діабетичну ретинопатію неможливо вилікувати, важливо виявити її на ранніх стадіях, щоб запобігти її подальшому розвитку. Наприклад, стадії непроліферативної діабетичної ретинопатії майже завжди міститимуть ранні ознаки діабетичної ретинопатії, і здатність виявити та класифікувати ці стадії за допомогою належної техніки оцінки може означати збереження зору.

Результати багатьох досліджень показують багатообіцяючу ефективність класифікації в цілому. Системи скринінгу, які розробляються сьогодні, можуть включати підходи, засновані на глибокому навчанні, для покращення та класифікації стадії діабетичної ретинопатії за допомогою методів виявлення уражень на кількох зображеннях очного дна. Основною проблемою є ручна діагностика, яка має відбуватися після скринінгу, що зазвичай є тривалим процесом, схильним до упередженості офтальмологів. Крім того, обмеження набору даних обмежують варіації зображення очного дна, які можна використовувати для оцінки показників.

Скринінг діабетичної ретинопатії залишається відкритим питанням через обмежену кількість загальнодоступних наборів даних, і хоча більшість останніх досягнень глибокого навчання досягають багатообіцяючих результатів у класифікації, деяким досі не вистачає здатності розрізняти уражені ураження. Інші методи просто ігнорують 5 стадій діабетичної ретинопатії, які вважаються ключовими для визначення тяжкості захворювання. Така відмінність у техніках

розкриває ще одну проблему. Той факт, що немає стандартних практик, які погоджують загальний набір стадій діабетичної ретинопатії, показує, що дослідники в цій галузі все ще можуть мати різні думки щодо достовірності результатів. Більшість кінцевих результатів, однак, можуть бути використані лише для діагностики та ніколи не вважаються остаточними.

Завдяки ефективності методів глибокого навчання аналіз сканів сітківки став швидшим, більш інклюзивним і доступним для узагальнення, однак показники, які використовуються для оцінки результатів, і відповідні набори даних залишаються упередженими та незбалансованими в різних дослідженнях. Зрештою, класифікація діабетичної ретинопатії має вирішальне значення, але розуміння різних причин також може бути важливою можливістю для дослідження. Наприклад, специфічні зміни ураження та інші приховані показники потенційно можуть вказувати на можливість розвитку діабетичної ретинопатії. Інші напрямки дослідження можуть включати вивчення діабетичного макулярного набряку, оскільки виявлення діабетичного макулярного набряку з високою ймовірністю означає, що на сітківці розвивається діабетична ретинопатія. Завдяки цим досягненням стало можливим узагальнити моделі, засновані на глибокому навчанні, і оцінити ширший спектр симптомів і показників, які могли б допомогти дослідникам краще зрозуміти причини захворювань сітківки.

РОЗДІЛ 3

МЕТОД МУЛЬТИМОДАЛЬНОГО АНАЛІЗУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ

За результатами проведеного дослідження та згідно поставленого завдання було запропоновано комплексний метод мультимодального аналізу медичних даних на основі глибокого навчання, що включає в себе:

- дослідження доповнення метаданих для вирішення завдання багатокласової класифікації [98];
- дослідження впливу складності мультимодальної моделі на вирішення завдання багатокласової класифікації [99];
- дослідження недетермінованого доповнення метаданих для вирішення завдання багатокласової класифікації [100];
- дослідження впливу мультимодального доповнення метаданих на точність багатокласової класифікації.

3.1. Доповнення метаданих для вирішення завдання багатокласової класифікації

На основі набору даних RetinaMNIST розглядається проблема багатокласової класифікації тяжкості діабетичної ретинопатії для одномодальної (з введенням зображення) моделі та мультимодальної (з введенням зображення та тексту) моделі. Вплив додаткових даних, таких як суб'єктивна думка “пацієнта” або думка “експерта” про стан здоров'я пацієнта (що забезпечує “витік даних” на деякі класи), може бути корисним у деяких практичних ситуаціях. Ці думки були змодельовані за допомогою додаткових (доповнених) метаданих із змодельованих анкет. У результаті були підготовлені наступні варіанти вхідних значень і відповідних моделей: одномодальна модель (SM) лише з вхідними зображеннями та мультимодальні моделі з вхідними зображеннями та текстом, наприклад мультимодальна модель з думкою пацієнта (MP), мультимодальна модель з думкою

експерта (ME) і мультимодальна модель з думкою пацієнта та експерта (MPE). Усі ці мультимодальні моделі (MP, ME, MPE) дозволили досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC для всіх класів діабетичної ретинопатії. Далі під терміном “модель” (або “модель нейронної мережі”) мається на увазі конкретна конфігурація вагів в нейронній мережі певної архітектури, наприклад, тренувана модель означає конфігурацію вагів в нейронній мережі певної архітектури після певного процесу тренування.

3.1.1. Набір даних

Фактично, для цілей цієї роботи набір даних RetinaMNIST (як частина MedMNIST) базується на виклику DeepDRiD, який надає набір даних із 1600 зображень очного дна сітківки [94].



Рисунок 3.1 – Приклади зображень для задачі багатокласової класифікації діабетичної ретинопатії з 5 класами

Зображення позначені 5-рівневою класифікацією тяжкості діабетичної ретинопатії, де медичні деталі їх діагностики наведені в іншому місці [94]. Вихідний набір для навчання був розділений у співвідношенні 9:1 на набір для навчання та перевірки, а вихідний набір для перевірки використовувався як

тестовий набір. Це співвідношення було використано для порівняння поточних результатів із попередніми [95, 96].

Набір даних складається з 1600 зображень очного дна сітківки, які розділені на 5 типів з огляду на спостережуваний рівень діабетичної ретинопатії [94]: відсутність видимої ДР – клас 0 (рис. 3.1.a), легка непроліферативна ДР – клас 1 (рис. 3.1.b), помірна непроліферативна ДР – клас 2 (рис. 3.1.c), важка непроліферативна ДР – клас 3 (рис. 3.1.d) і проліферативна ДР – клас 4 (рис. 3.1.e).

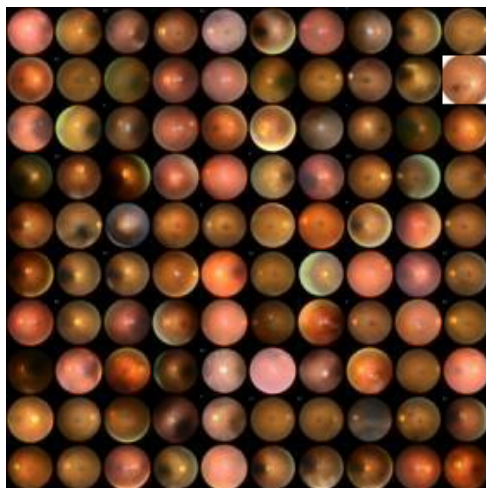


Рисунок 3.2 – Приклади зображень для задачі багатокласової (5 класів) класифікації в наборі даних MedMNIST із розмірами зображень 28x28

Вихідні зображення з великим розміром $3 \times 1736 \times 1824$ були обрізані по центру та змінені до $3 \times 28 \times 28$ (рис. 3.2). Усі ці зображення були попередньо оброблені з маленького розміру $3 \times 28 \times 28$ до трохи більшого розміру $3 \times 32 \times 32$, через вимоги стандартних архітектур нейронних мереж, що використовуються. Нарешті, набір даних RetinaMNIST (як частина MedMNIST) містить частини навчання (1080 зображень), перевірки (120 зображень) і тесту (400 зображень).

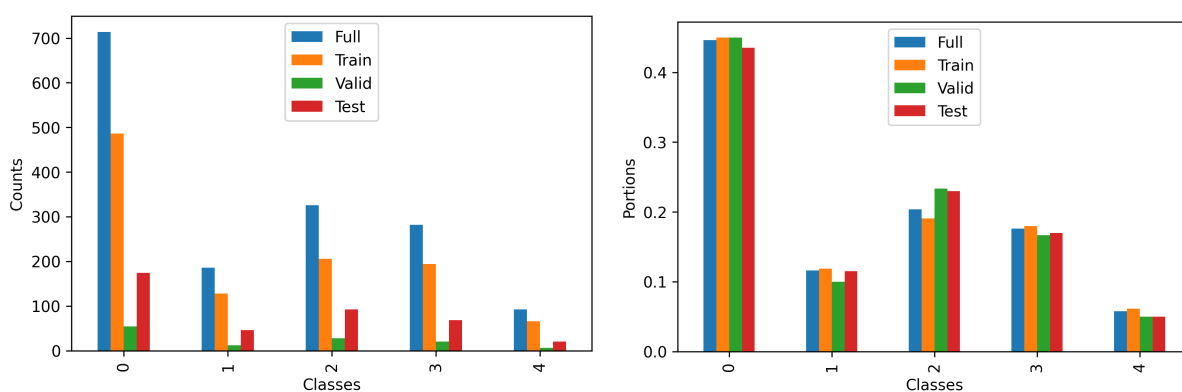


Рис. 3.3. – Розподіл зображень за класами в кількості і пропорціях

Дослідницький аналіз даних був виконаний для розуміння представлення даних за класами. Наведено розподіл зображень за класами в кількості і пропорціях (рис. 3.3), де помітна виражена зміщеність. У майбутньому, щоб уникнути такої незбалансованої конфігурації, слід застосовувати ваги класів, адекватні фактичному представленню зображень класами. Слід зазначити, що такий самий розподіл зображень за класами зберігся для підмножин набору даних для навчання, перевірки та тестування (рис. 3.3).

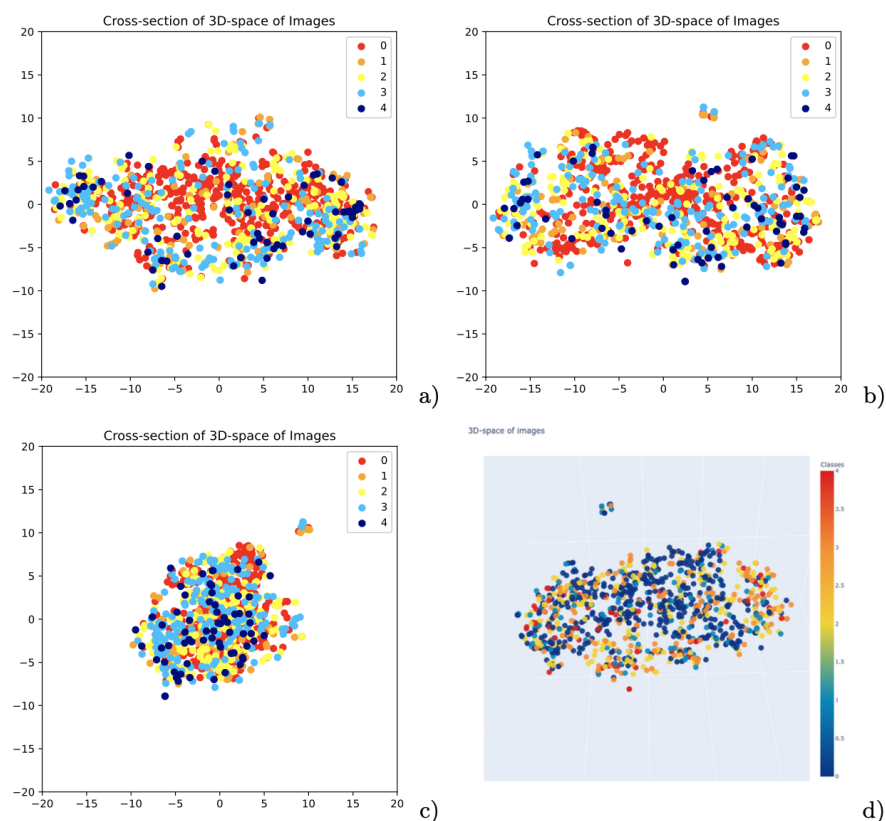


Рисунок 3.4 – Візуалізація моделі t-SNE для зображень сітківки за допомогою тривимірної точки

Для візуалізації подібності та відмінності зображень сітківки для різних класів було застосовано відомий метод t-розподіленого стохастичного вбудовування сусідів (t-SNE) [91]. Це дозволило вбудувати високовимірні дані зображення сітківки в 3D-простір, який потім можна візуалізувати на діаграмі розсіювання, де подібні зображення сітківки моделюються сусідніми точками, а відмінні зображення сітківки відображаються на віддалених точках: а) XY-проекція, б) XZ-проекція, в) YZ-проекція, г) 3D представлення (рис. 3.4). Власне, ця карта t-SNE зображень сітківки в 3D-просторі дозволяє візуально зрозуміти дуже близьку

схожість об'єктів з різних класів, які навряд чи можна розрізнити простими підходами. Слід підкреслити, що незважаючи на досить низьку візуальну якість спрощеного набору даних RetinaMNIST (32x32) (рис. 3.2) порівняно з оригінальним набором даних DeepDRiD (1736x1824) (рис. 3.1), деякі класи, такі як клас 0 (рис. 3.4, червоний колір) і клас 4 (рис. 3.4, темно-синій колір) мають візуально виділені області розташування. У той же час, інші класи (наприклад, клас 1, клас 2, клас 3) важко розрізнити, і для вирішення завдання багатокласової класифікації діабетичної ретинопатії слід використовувати більш складні підходи на основі глибокого навчання.

В деяких практичних ситуаціях може бути корисним вплив додаткових даних, таких як суб'єктивна думка “пацієнта” або думка “експерта” про стан здоров'я пацієнта. Ці думки були змодельовані за допомогою додаткових (доповнених) метаданих із змодельованих анкет, щоб отримати:

- думку “пацієнта” як відповідь на питання «Чи відчуваєтеся Ви здоровим?» з 2 можливими відповідями: “здоровий” (англ. *healthy*, для класу 0) або “хворий” (англ. *ill*, для класів від 1 до 4), де будь-який дискомфорт пацієнта приблизно класифікується як “хворий” стан,

- думку “експерта” як відповідь на питання «Чи бачите Ви важкий хворобливий стан?» з 2 можливими відповідями: “хворий” (англ. *ill*, для класів від 0 до 3) або “важкий” (англ. *severe*, для класу 4), де будь-який неважкий проліферативний стан приблизно класифікується як “хворий”.

Загалом, думки “пацієнта” та “експерта” можуть бути набагато складнішими і містити схожі слова навіть із різною семантикою. У цій роботі було надзвичайно спрощено зрозуміти вплив додаткової модальності на показники процедури глибокого навчання. У майбутньому більш складні текстові висновки можуть бути використані для вивчення такого впливу з більш реалістичних точок зору.

В результаті були підготовлені наступні варіанти вхідних значень (табл. 3.1):

- одномодальна модель (SM) – вхідні дані складаються лише з зображень,
- мультимодальна модель з думкою пацієнта (MP) – вхідні дані складаються із зображень і тексту думки пацієнта, як-от “здоровий” або “хворий”,

- мультимодальна модель з думкою експерта (ME) – вхідні дані складаються із зображень і тексту думки експерта, а саме “ill” або “severe”,
- мультимодальна модель з думкою пацієнта і експерта (MPE) – вхідні дані складаються із зображень і тексту думки пацієнта і експерта, а саме “healthy” або “ill” і “ill” або “severe”.

Таблиця 3.1 – Вхідні значення для моделі глибокого навчання

Модель	Вхідні дані	Клас 0	Клас 1	Клас 2	Клас 3	Клас 4
SM	input_image	Рис. 3.1.a	Рис. 3.1.b	Рис. 3.1.c	Рис. 3.1.d	Рис. 3.1.e
MP	input_image + input_text	Рис. 3.1.a + healthy	Рис. 3.1.b + ill	Рис. 3.1.c + ill	Рис. 3.1.d + ill	Рис. 3.1.e + ill
ME	input_image + input_text	Рис. 3.1.a + ill	Рис. 3.1.b + ill	Рис. 3.1.c + ill	Рис. 3.1.d + ill	Рис. 3.1.e + severe
MPE	input_image + input_text	Рис. 3.1.a + healthy + ill	Рис. 3.1.b + ill + ill	Рис. 3.1.c + ill + ill	Рис. 3.1.d + ill + ill	Рис. 3.1.e + ill + severe

3.1.2. Моделі

Творці MedMNIST порівняли кілька стандартних методів глибокого навчання та інструментів AutoML, включаючи глибокі нейронні мережі (DNN) з архітектурою ResNet зі стратегіями ранньої зупинки на наборі даних перевірки. Для перевірки можливості практичного використання архітектури ResNet було виконано експериментальне дослідження і експериментально доведено ефективність використання нейронних мереж вказаних архітектур (ResNet, WRN, VGG) за умов значного зменшення розміру мережі на основі дистиляції знань [101]. Тут використано кілька архітектур глибокої нейронної мережі для роботи з вхідними даними різних модальностей (табл. 3.1) на основі стандартної згорткової нейронної мережі для обробки вхідних даних зображень на основі ResNet50 (для порівняння з попередніми результатами [95, 96]) і рекурентної нейронної мережі (RNN) для обробки вхідних даних тексту:

- одна модальність (модель SM) – згорткова нейронна мережа з

архітектурою ResNet50 для обробки вхідних даних зображень (рис. 3.5.a),

- мультимодальність (моделі MP, ME, MPE) – згортова нейронна мережа з архітектурою ResNet50 для обробки вхідних даних зображень та рекурентна нейронна мережа з архітектурою LSTM (рис. 3.5.b) для обробки вхідних даних тексту.

Варта зазначити, що в проведених в рамках даної роботи дослідженнях вхідними текстовими даними є максимально спрощені дані, які можна обробляти більш простими компонентами, проте тут використана рекурентна нейронна мережа з архітектурою LSTM, щоб показати, що дане дослідження націлене на розвиток в майбутньому для обробки складних текстових даних, для яких використовується дана мережа.

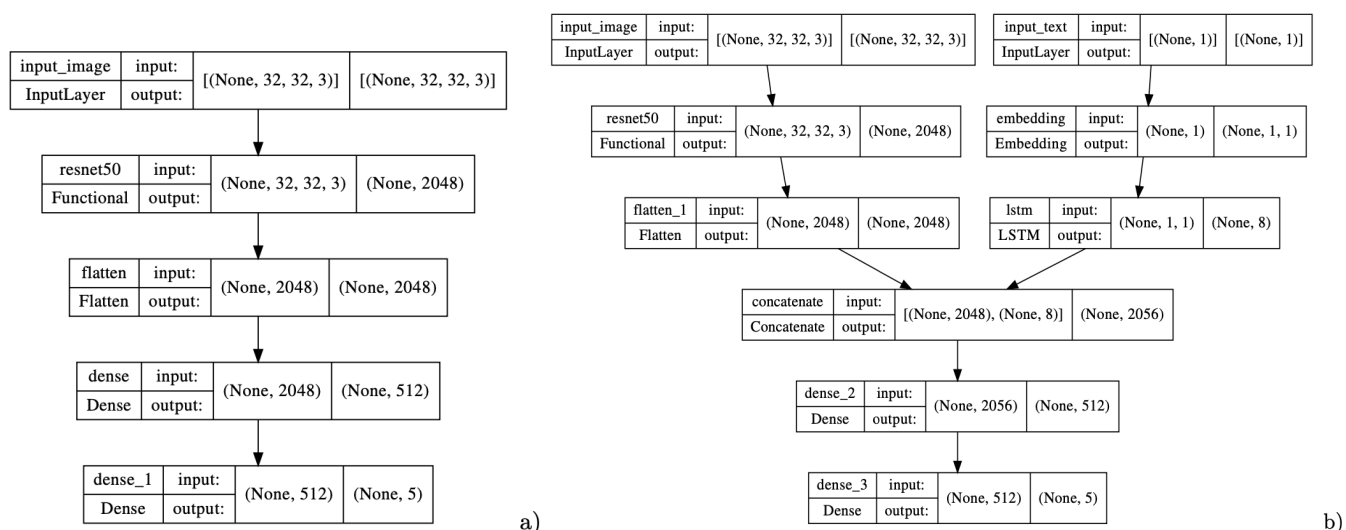


Рисунок 3.5 – Візуалізація використаних моделей

Для порівняння мультимодальних підходів, запропонованих тут, із порівняльними результатами проблеми багатокласової класифікації [95, 96] у роботі використовувалася популярна стандартна модель DNN з архітектурою ResNet, а саме ResNet50. Стандартну архітектуру використовуваних DNN було частково змінено з урахуванням кількості використовуваних класів, а саме: видалено останній шар класифікації, додано щільний шар із 512 вузлами, додано щільний шар класифікації з 3 вузлами, який визначає остаточну категорію зображення. Моделі DNN тренувалися з початковими випадковими вагами протягом 100 епох, тому що, як виявилось, цього було достатньо для насичення зростання метрики

перевірки в багатьох випадках. Оптимізація проводилась за алгоритмом Адама зі швидкістю навчання, яка дорівнювала 0,001, експоненціальною швидкістю згасання для 1-го моменту – 0,9 та експоненціальною швидкістю згасання для 2-го моменту – 0,99, і константою епсилон – 10^{-7} [92]. Для порівняння з попередніми результатами було обрано розмір пакету для початкових прогонів 128 [95, 96]. Точність, AUC і втрати були обрані в якості основних показників.

3.1.3. Робочий процес

Весь робочий процес було реалізовано як режим перехресної перевірки [93] та подальший режим поза межами перехресної перевірки, де:

- режим перехресної перевірки (CV) виконувався на основі навчальної частини (1080 зображень) оригінального набору даних RetinaMNIST, який був розділений на шість частин, де найкраща модель після навчання на частині CV-навчання (900 зображень) була визначена за значенням AUC після валідації на частині CV-валідації (180 зображень) для кожної з шести частин, потім метрики були виміряні для кожної з частин на частині для тестування оригінального набору даних RetinaMNIST (400 зображень), а їхнє середнє та стандартне відхилення було розраховано для розуміння статистичної достовірності отриманих даних;
- режим поза межами перехресної перевірки (OoCV) був реалізований шляхом навчання на всій частині для навчання (1080 зображень) вихідного набору даних RetinaMNIST, шляхом вибору найкращої моделі за значенням AUC після перевірки на частині для валідації (120 зображень) оригінального набору даних RetinaMNIST, і остаточного тестування на частині для тестування (400 зображень) оригінального набору даних RetinaMNIST.

3.1.4. Програмні засоби

Для проведення дослідження було створено програмний засіб, який виконує наступний алгоритм:

1. завантаження набору даних;
2. попередня обробка набору даних;
3. створення текстової модальності;
4. створення моделі нейронної мережі;
5. тренування та тестування моделі в режимі CV;
6. тренування та тестування моделі в режимі OoCV;
7. опрацювання отриманих результатів;
8. візуалізація отриманих результатів.

Для реалізації даного алгоритму були створені наступні програмні компоненти:

- компонента обробки вхідних даних (код реалізації наведено в додатку А, лістинг А.1);
- компонента для роботи нейронної мережі (код реалізації наведено в додатку А, лістинг А.2);
- компонента обробки результатів (код реалізації наведено в додатку А, лістинг А.3).

3.2. Вплив складності мультимодальної моделі на вирішення завдання багатокласової класифікації

Проблема багатокласової класифікації тяжкості діабетичної ретинопатії розглядається на основі набору даних RetinaMNIST за допомогою одномодальних (з введенням зображення) і багатомодальних (з введенням зображення та тексту) моделей DNN. На практиці додаткові дані, такі як суб'єктивна думка “пацієнта” про свій стан здоров'я, які спричиняють “витік даних” щодо конкретних класів, можуть бути корисними. Для імітації цих думок використовувались додаткові (доповнені) метадані змодельованих опитувань. У результаті мультимодальні моделі дозволили досягти статистично значущих покращень точності багатокласової класифікації за значенням AUC для деяких класів, ці покращення виходять за межі стандартного відхилення, вимірюного перехресною перевіркою, і можуть бути оцінені як значні.

3.2.1. Набір даних

Для цього дослідження використовувався набір даних RetinaMNIST, який детально описано в розділі 3.1.1. У RetinaMNIST вихідні зображення розміру $3 \times 1736 \times 1824$ були обрізані по центру та зменшені до розміру $3 \times 28 \times 28$ (рис. 3.2). Для використання зображень зі стандартними згортковими нейронними мережами зображення попередньо оброблено з формату $3 \times 28 \times 28$ до $3 \times 32 \times 32$.

Також використовувались додаткові дані, такі як суб'єктивна думка “пацієнта”, опис моделювання якої детально описано в розділі 3.1.1.

Таблиця 3.2. Вхідні значення для моделі глибокого навчання

Модель	Вхідні дані	Клас 0	Клас 1	Клас 2	Клас 3	Клас 4
SM	input_image	Рис. 3.1.a	Рис. 3.1.b	Рис. 3.1.c	Рис. 3.1.d	Рис. 3.1.e
MP	input_image + input_text	Рис. 3.1.a + healthy	Рис. 3.1.b + ill	Рис. 3.1.c + ill	Рис. 3.1.d + ill	Рис. 3.1.e + ill

В результаті були підготовлені наступні варіанти вхідних значень (табл. 3.2):

- одномодальна модель (SM) – вхідні дані складаються лише з зображень,
- мультимодальна модель з думкою пацієнта (MP) – вхідні дані складаються із зображень і тексту думки пацієнта, а саме “healthy” або “ill”.

3.2.2. Моделі

Тут було використано кілька архітектур глибокої нейронної мережі для роботи з вхідними даними різних модальностей (табл. 3.2) на основі стандартної згорткової нейронної мережі (табл. 3.3) для обробки вхідних даних зображень і рекурентної нейронної мережі для обробки вхідних даних тексту:

- одна модальність (модель SM) – різні види архітектури згорткової нейронної мережі для обробки вхідних даних зображень,
- мультимодальність (модель MP) – різні види архітектури згорткової нейронної мережі для обробки вхідних даних зображень та рекурентна нейронна мережа з архітектурою LSTM для обробки вхідних даних тексту.

Таблиця 3.3 – Архітектури глибокої нейронної мережі

Модель	CNN		RNN
SM	ResNet50, MobileNetV2, DenseNet201, NASNetLarge, EfficientNetB7, VGG19, EfficientNetV2L, NASNetMobile, EfficientNetB0, EfficientNetV2S	+	–
MP			LSTM

Для порівняння підходів, запропонованих в цьому дослідженні, із еталонними результатами, особливо на ResNet50, у роботі використовувалися різні види архітектури CNN. Кілька класичних нейронних мереж використовувалися для компоненти CNN архітектури DNN (табл. 3.3): VGG, ResNet50, MobileNetV2, DenseNet201, NASNet і EfficientNet. Загальна архітектура компоненти CNN (табл. 3.3), яка детально описана в розділі 3.1.2, була застосована до інших видів компоненти CNN, які використовувалися в цьому дослідженні. Основними показниками для аналізу отриманих результатів були обрані AUC і кількість параметрів у компонентах CNN.

3.2.3. Робочий процес

Загалом схема робочого процесу (додаток Г) включає в себе дослідницький аналіз даних, підготовку даних, тренування і тестування моделі та перевірку математичного опису всі ці етапи робочого процесу реалізовано за допомогою програмних засобів, які створюються в даній роботі. Далі під терміном “математичний опис” мається на увазі теоретичне обґрунтування кількісної оцінки і якісної інтерпретації експериментально знайденого підвищення точності багатокласової класифікації. Проведений дослідницький аналіз даних детально описано в розділі 3.1.1. Підбір модальностей для кожного дослідження описано в розділах 3.1.1, 3.2.1, 3.3.2, 3.4.2.

Далі робочий процес було реалізовано як режим CV із подальшим режимом OoCV (шляхом навчання на всій частині для навчання (1080 зображень) вихідного набору даних RetinaMNIST, вибору найкращої моделі за значенням AUC після перевірки на частині для перевірки (120 зображень) вихідного набору даних

RetinaMNIST і остаточного тестування на частині для тестування (400 зображень) вихідного набору даних RetinaMNIST), який детально описано в розділі 3.1.3.

Етап перевірки математичного опису, який наведений на схемі, використовується та детально описаний в розділі 3.4.

3.2.4. Програмні засоби

Для проведення даного дослідження використовувалися програмні засоби з дослідження доповнення метаданих для вирішення завдання багатокласової класифікації з єдиною зміною у компоненті для роботи нейронної мережі. Оскільки в цьому дослідженні розглядались різні види архітектури згорткової нейронної мережі для обробки вхідних даних зображень, то в програмній реалізації згаданої компоненти нейронна мережа з архітектурою ResNet50 була замінена на нейронну мережу з архітектурами MobileNetV2, DenseNet201, NASNetLarge, EfficientNetB7, VGG19, EfficientNetV2L, NASNetMobile, EfficientNetB0, EfficientNetV2S відповідно.

3.3. Недетерміноване доповнення метаданих для вирішення завдання багатокласової класифікації

Тут запропоновано метод доповнення недетермінованих метаданих (англ. fuzzy metadata augmentation – FMDA) для задачі багатокласової класифікації зображень для одномодальної моделі (із введенням зображення) та мультимодальної моделі (із введенням зображення та тексту). Вплив додаткових даних, таких як суб’єктивна недетермінована думка “експерта” про стан здоров’я пацієнта (які забезпечують “витік даних” для деяких екстремальних та подібних класів), може бути корисним у деяких практичних ситуаціях. Ця думка була змодельована за допомогою додаткових (доповнених) метаданих із змодельованих анкет для стандартних CIFAR10/CIFAR100 і спеціалізованих наборів даних MedMNIST. Були підготовлені наступні варіанти вхідних даних і відповідних

моделей: одномодальна модель (SM) лише з вхідними зображеннями та мультимодальна модель з недетермінованою думкою експерта (MMFE) із вхідними зображеннями та текстом думки експерта. Модель MMFE дозволила досягти різних статистично значущих покращень точності багатокласової класифікації за значеннями AUC для всіх класів. Загалом, FMDA може бути корисною стратегією для кращої класифікації деяких важко класифікованих захворювань та в більш загальному контексті.

3.3.1. Набори даних

В цьому дослідженні були використані стандартні набори даних CIFAR10/CIFAR100 і спеціалізовані (медичні) набори даних MedMNIST.

Стандартні CIFAR10 і CIFAR100 є промаркованими підмножинами (названими на честь Канадського інституту перспективних досліджень) набору даних багатьох малих зображень [89]. Набір даних CIFAR10 – це набір зображень, які широко використовувалися для тестування різних алгоритмів комп’ютерного зору, машинного навчання і глибокого навчання. Він містить 60 000 кольорових зображень у 10 різних класах із розміром зображення 32x32 пікселя (рис. 3.6.a). До 10 різних класів входять літаки, автомобілі, птахи, коти, олені, собаки, жаби, коні, кораблі та вантажівки з 6000 зображеннями на кожен клас. Набір даних CIFAR100 подібний до набору даних CIFAR10, але містить зображення 100 класів із 600 зображеннями на клас (рис. 3.6.b). 100 класів у CIFAR100 були згруповані в 20 так званих суперкласів, де кожне зображення має мітку “fine” (для класу, до якого воно належить) і мітку “coarse” (для суперкласу, до якого воно належить).

Набір даних MedMNIST v2 – це великомасштабний супернабір даних, подібний до MNIST, із стандартизованими 2D та 3D біомедичними даними [95, 96]. Усі зображення були попередньо оброблені до невеликого розміру 28×28 (2D) або 28×28×28 (3D) з відповідними класифікаційними мітками. Вони включають деякі базові модальності даних у біомедичних зображеннях і призначені для виконання класифікації легких 2D і 3D зображень із набором даних різного обсягу (від 100 до

100 000 зображень) і різними завданнями (наприклад, для бінарної або багатокласової класифікації). Ці набори даних використовувалися в різних медичних і загальнодослідницьких освітніх цілях у біомедичних комп'ютерних системах зору, машинному та глибокому навчанні. Тут використовуються такі набори даних із колекції MedMNIST: BloodMNIST із зображеннями аналізу крові (рис. 3.6.c), DermaMNIST із дерматоскопічними зображеннями (рис. 3.6.d), PathMNIST із гістологічними зображеннями (рис. 3.6.e) та RetinaMNIST із зображеннями очного дна сітківки (рис. 3.6.f).

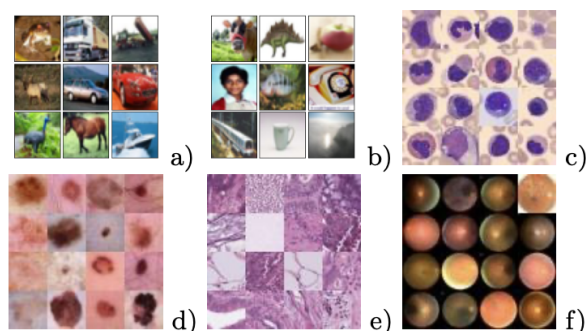


Рисунок 3.6 – Приклади зображень із наборів даних, які використовувалися в дослідженні

Набір даних BloodMNIST базується на оригінальному наборі даних [84], який містить зображення окремих нормальних клітин, отриманих від осіб без інфекцій або захворювань і без будь-якого фармакологічного лікування на момент забору крові. Він містить понад 17 000 зображень 8 класів. Вихідні зображення розміром $3 \times 360 \times 363$ були обрізані по центру до розміру $3 \times 200 \times 200$, а потім масштабовані в зображення розміром $3 \times 28 \times 28$ (рис. 3.6.c).

Набір даних DermaMNIST базується на HAM10000 [81, 82], який містить дерматоскопічні зображення пігментованих уражень шкіри. Він містить понад 10 000 дерматоскопічних зображень з 7 різних класів. Вихідні зображення розміром $3 \times 600 \times 450$ були масштабовані в зображення розміром $3 \times 28 \times 28$ (рис. 3.6.d).

Набір даних PathMNIST базується на наборі даних NCT-CRC-HE-100K після дослідження для прогнозування виживаності на слайдах гістології колоректального раку для більш ніж 100 000 ділянок зображень із пофарбованих гістологічних зображень, а також тестового набору даних (CRC-VAL-HE-7K) з більше ніж 7000

патчів зображень з різних клінічних центрів [106]. Набір даних містить зображення 9 типів класів. Вихідні зображення розміром $3 \times 224 \times 224$ були масштабовані на зображення розміром $3 \times 28 \times 28$ (рис. 3.6.e).

Набір даних RetinaMNIST базується на наборі даних із виклику DeepDRiD [103]. Він містить понад 1500 зображень очного дна сітківки 5 класів для 5-рівневої оцінки тяжкості діабетичної ретинопатії. Вихідні зображення розміром $3 \times 1736 \times 1824$ були обрізані по центру з розміром вікна, що дорівнює довжині короткого краю, а потім масштабовано до розміру $3 \times 28 \times 28$ (рис. 3.6.f).

3.3.2. Моделі і доповнення недетермінованих метаданих

У цьому дослідженні була розширена згадана раніше модель за допомогою недетермінованого доповнення метаданих (FMDA). У результаті були побудовані моделі на основі ResNet: одномодальна модель (SM) лише з вхідними зображеннями (рис. 3.5.a) і мультимодальна модель з недетермінованою думкою експерта (MFE) із вхідними зображеннями та недетермінованим текстом думки експерта (рис. 3.5.b). Значення AUC було обране як основний показник для порівняння еталонного результату з результатами, отриманими в цьому дослідженні. Перехресна перевірка була виконана для всіх моделей щодо підмножини для тренування стандартних і спеціалізованих наборів даних, згаданих в розділі 3.3.1.

Моделі MFE були підготовлені відповідно до наступних моделей FMDA. Наприклад, шаблон F_N (“одна детермінована думка експерта”) означає одну додаткову текстову модальність з точним маркуванням для класу N , наприклад, $N = 0$ для класу 0 (“здоровий” для RetinaMNIST) та недетерміноване маркування для всіх інших класів $N-1$ (“хворий” для RetinaMNIST з вищими рівнями тяжкості захворювання).

Більш складні шаблони F_k_l або $F_k_l_m$ (“кілька недетермінованих думок експерта”) означають групи класів k , l і m , які були промарковані загальною недетермінованою міткою як нерозрізнені класи. Наприклад, “ $F_1_2_2$ ” означає 1

додаткову текстову модальність з одним точним маркуванням для 1-го класу (значення “1” для класу 0 – “здоровий”), недетерміноване маркування для наступних 2 класів (значення “2” для класів від 1 до 2 – “хворий” з різними ступенями тяжкості), і наступне недетерміноване маркування для наступних 2 класів (значення “2” для класів від 3 до 4 – “хворий” з вищими рівнями тяжкості).

Окремий шаблон `F_super20`, який використовується лише для набору даних CIFAR100, означає, що кожен клас із 20 суперкласів отримав додаткову текстову мітку цього суперкласу, яка створила додаткове недетерміноване маркування кожного класу з поточного суперкласу. Це призводить до додаткових 20 недетермінованих суперміток, тобто 1 недетермінованої супермітки для 5 класів у кожному суперкласі.

3.3.3. Програмні засоби

Для проведення даного дослідження використовувалися програмні засоби з дослідження доповнення метаданих для вирішення завдання багатокласової класифікації з деякими змінами, а саме:

- у компоненті обробки вхідних даних був модифікований етап створення текстової модальності згідно з шаблонами описаними в розділі 3.3.2 (фрагменти коду реалізації наведено в додатку Б, лістинг Б.1).

3.4. Вплив мультимодального доповнення метаданих на точність багатокласової класифікації

3.4.1. Набори даних

Використані набори даних включають: CIFAR10 з різними зображеннями загального призначення (рис. 3.7.a), PathMNIST з гістологічними зображеннями (рис. 3.7.b) і RetinaMNIST із зображеннями очного дна сітківки (рис. 3.7.c).

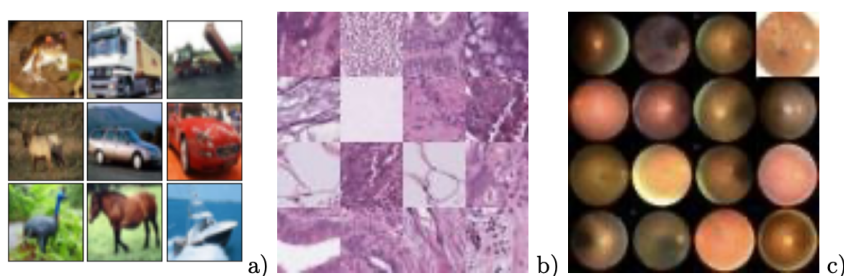


Рисунок 3.7 – Приклади зображень із наборів даних, які використовувалися в дослідженні

3.4.2. Моделі теоретичної оцінки впливу мультимодального доповнення метаданих на точність багатокласової класифікації

У практичних сценаріях включення додаткових даних, таких як суб'єктивна оцінка пацієнтами стану свого здоров'я, може дати явні переваги. Ці суб'єктивні вхідні дані були імітовані шляхом інтеграції доповнених метаданих, отриманих із змодельованих анкет. Ця інтеграція мала на меті отримати форму “часткової” думки, подібної до думки пацієнта чи експерта, у відповідь на запитання «Який із цих класів є класом i ?» Для кожного класу було дві можливі відповіді: “цей” (для класу i) або “інший” (для класів, відмінних від класу i). Цей підхід є узагальненням суб'єктивних думок, пов'язаних з поточним класом. На практиці часткові думки можуть бути досить заплутаними. Однак у цьому дослідженні ця концепція була спрощена, щоб дослідити, як додаткові модальності впливають на показники глибокого навчання. У результаті були підготовлені наступні варіанти вхідних значень (табл. 3.4): одномодальна модель (SM) – вхідні дані складаються лише із зображення, мультимодальна модель з думкою пацієнта (MP) – вхідні дані складаються із зображення та тексту часткової думки, а саме маркерів “цей клас i ” або “інші класи (всі інші, крім i)”.

В дослідженні було використано кілька моделей для роботи з вхідними даними різної модальності (табл. 3.4) на основі стандартної згорткової нейронної мережі ResNet50 для обробки вхідних даних зображень та рекурентної нейронної мережі LSTM для обробки текстових вхідних даних (табл. 3.5). Архітектура

моделей нейронних мереж детально описана в дослідженні доповнення метаданих для вирішення завдання багатокласової класифікації в розділі 3.1.2.

Таблиця 3.4 – Вхідні значення для моделі глибокого навчання

Модель	Вхідні дані	Клас i	Інші класи (всі, крім i)
SM	input_image	image	image
MP	input_image + input_text	image + “this”	image + “other”

Таблиця 3.5 – Архітектура глибокої нейронної мережі

Модель	CNN		RNN
SM	ResNet50	+	–
MP			LSTM

Площа під кривою (AUC), точність за класом i усереднена точність були обрані як основні показники для аналізу отриманих результатів. Весь робочий процес було реалізовано як дослідження перехресної перевірки (CV) з обчисленням середнього значення та стандартних відхилень показників, які використовувалися та детально описані в дослідженні впливу складності мультимодальної моделі на вирішення завдання багатокласової класифікації.

3.4.3. Програмні засоби

Для проведення даного дослідження використовувалися програмні засоби з дослідження доповнення метаданих для вирішення завдання багатокласової класифікації з деякими змінами, а саме:

- у компоненті обробки вхідних даних був модифікований етап створення текстової модальності згідно з описом в розділі 3.4.2 (фрагменти коду реалізації наведено в додатку В, лістинг В.1);
- у компоненті для роботи нейронної мережі було усунено етап тренування та тестування моделі в режимі OoCV, оскільки в даному дослідженні не використовуються результати цього етапу;

- у компоненті обробки результатів модифіковано етапи опрацювання та візуалізації отриманих результатів згідно з цілями даного дослідження (фрагменти коду реалізації наведено в додатку В, лістинг В.2).

3.4.4. Матриця невідповідностей: Табличний і матричний вигляд

Для стандартної матриці невідповідностей для багатокласової задачі з n класами та значеннями хибно позитивних (FP), хибно негативних (FN) і істинно позитивних (TP) результатів у вигляді таблиці (табл. 3.6) можна використовувати відповідний матричний запис у вигляді рівняння (1). Кожен рядок матриці представляє екземпляри в фактичному класі, тоді як кожен стовпець представляє екземпляри в передбаченому класі, де всі значення виражені у відносних одиницях у діапазоні $[0, 1]$ для позначення частини екземплярів кожного класу.

Варто зазначити, що тут і далі використовується матриця невідповідностей саме для багатокласової класифікації, тому така матриця не містить істинно негативних (TN) результатів. Істинно негативні результати доцільно розглядати для матриці невідповідностей для бінарної класифікації або для матриці невідповідностей для кожного окремого класу при багатокласовій класифікації, тоді такі результати з'являтимуться в діагональних елементах матриці, які не лежать на перетині рядка і стовпця фактичного і передбаченого конкретного класу.

Таблиця 3.6 – Матриця невідповідностей для класифікації з n класами

		Передбачений				
		Клас 0	...	Клас j	...	Клас $n-1$
Фактичний	Клас 0	$TP_{0,0}$...	$FN_{0,j}$...	$FN_{0,n-1}$

	Клас j	$FP_{i,0}$...	$TP_{i,j}$...	$FN_{i,n-1}$

	Клас $n-1$	$FP_{n-1,0}$...	$FP_{n-1,j}$...	$TP_{n-1,n-1}$

Матрицю невідповідностей для вхідних даних однієї модальності (C_s) у матричному вигляді буде подано як (1), де $a_{i,j} = TP_{i,j}$, $n_{i,j} = FN_{i,j}$, $p_{i,j} = FP_{i,j}$.

$$C_s = \begin{bmatrix} a_{0,0} & \dots & n_{0,j} & \dots & n_{0,n-1} \\ \dots & \dots & \dots & \dots & \dots \\ p_{i,0} & \dots & a_{i,j} & \dots & n_{i,n-1} \\ \dots & \dots & \dots & \dots & \dots \\ p_{n-1,0} & \dots & p_{n-1,j} & \dots & a_{n-1,n-1} \end{bmatrix} \quad (1)$$

3.4.5. Матриця невідповідностей: Властивості

Тут описано властивості матриці невідповідностей для задачі багатокласової класифікації, які є неочевидними і необхідні для виведення наступних результатів.

Лема 1. У матриці C_s сума всіх елементів у кожному k -му рядку дорівнює 1,

$$\sum_{j=0}^{k-1} p_{k,j} + a_{k,k} + \sum_{j=k+1}^{n-1} n_{k,j} = 1, \quad (2)$$

де $k \in [0, n - 1]$.

Доведення. Оскільки кожен елемент k -го рядка матриці C_s (1) представляє частину екземплярів фактичного k -го класу, то їхня сума вздовж k -го рядка дасть усю множину значень k -го класу. Тому ця сума елементів, виражена у відносних одиницях, буде дорівнювати 1.

Матриця невідповідностей для мультимодальних (зображення + мітка k) вхідних даних, а саме $CM_m(k)$, матиме таку форму (3):

$$CM_m(k) = \begin{bmatrix} a_{0,0} & \dots & n_{0,j=k} & \dots & n_{0,n-1} \\ \dots & \dots & \dots & \dots & \dots \\ p_{i,0} & \dots & a_{i=k,j=k} & \dots & n_{i,n-1} \\ \dots & \dots & \dots & \dots & \dots \\ p_{n-1,0} & \dots & p_{n-1,j=k} & \dots & a_{n-1,n-1} \end{bmatrix} \quad (3)$$

з властивостями, викладеними в лемі 2 нижче.

Лема 2. У матриці $CM_m(k)$:

- сума всіх елементів k -го рядка дорівнює 1;
- сума всіх елементів k -го стовпця дорівнює 1;
- для стовпців $j \in [0, n - 1], j \neq k, p_{k,j} = 0, n_{k,j} = 0$;
- для рядків $i \in [0, n - 1], i \neq k, p_{i,k} = 0, n_{i,k} = 0$;
- $a_{k,k} = 1$.

Доведення. Явне знання про цей клас k було визначено міткою k у мультимодальних вхідних даних. Ось чому немає невизначеності щодо передбачення k -го класу та $a_{k,k} = 1$. Ось чому всі хибно позитивні ($p_{k,j}$) та всі хибно негативні ($n_{k,j}$) для k -го класу (значення в k -му рядку, крім елемента з індексом (k, k)) дорівнюють 0 для $j \in [0, n - 1]$.

Також для всіх інших $i \neq k$ класів немає невизначеності щодо хибного передбачення i -го класу як k -го класу. Тому всі хибно негативні ($n_{i,k}$) та всі хибно позитивні ($p_{i,k}$) (значення в k -му стовпці) для k -го класу дорівнюють 0 для $i \in [0, n - 1]$.

Порівнюючи $C_s(1)$ і $C_m(k)$ (4), можна зробити висновок, що класифікаційні невизначеності k -го класу в $C_s(1)$ для кожного k -го рядка виражаються елементами $n_{k,j}$ і $p_{k,j}$, які згортаються до нульових значень із додаванням своїх значень до $a_{k,k}$, що дорівнює 1 згідно з лемою 1.

Згідно з лемою 2 матричний запис $CM_m(k)$ (3) матиме такий вигляд (4):

$$C_m(k) = \begin{bmatrix} a_{0,0} & \dots & 0 & \dots & n_{0,n-1} \\ \dots & \dots & 0 & \dots & \dots \\ 0 & 0 & a_{k,k} = 1 & 0 & 0 \\ \dots & \dots & 0 & \dots & \dots \\ p_{n-1,0} & \dots & 0 & \dots & a_{n-1,n-1} \end{bmatrix} \quad (4)$$

Лема 3 (Додавання недіагональних елементів k -го рядка до діагонального елемента $a_{k,k}$). Застосування мультимодального підходу призводить до перерозподілу значень у матриці невідповідностей від вигляду $C_s(1)$ до вигляду

$C_m(k)$ (4), а це призводить до додавання значень з недіагональних елементів k -го рядка у $C_s(1)$ до діагонального елемента $a_{k,k}$ у $C_m(k)$ (4).

Доведення. З леми 1 сума елементів у кожному рядку в $C_s(1)$ і в $C_m(k)$ (4) повинна бути постійною і дорівнювати 1.

З леми 2 у k -му рядку можна порівняти ненульові недіагональні елементи $n_{k,j}$ та $p_{k,j}$ із $C_s(1)$ та нульові недіагональні елементи $n_{k,j}$ та $p_{k,j}$ із $C_m(k)$ (4). Тоді можна вивести, що для збереження суми елементів у цьому k -му рядку, що дорівнює 1, ненульові недіагональні елементи із $C_s(1)$ фактично повинні бути агреговані в діагональний елемент $a_{k,k}$ в $C_m(k)$ (4).

Згідно з лемою 3 це призведе до:

$$a(k)_{k,k} = \sum_{j=0}^{k-1} p_{k,j} + a_{k,k} + \sum_{j=k+1}^{n-1} n_{k,j} = 1, \quad (5)$$

де k у дужках в $a(k)_{k,k}$ означає, що цей елемент $a(k)_{k,k}$ належить до $C_m(k)$ (4), а елемент $a_{k,k}$ належить до $C_s(1)$.

Лема 4 (Додавання недіагональних елементів k -го стовпця до інших елементів у відповідних рядках). Застосування мультимодального підходу призводить до перерозподілу значень у матриці невідповідностей від вигляду $C_s(1)$ до вигляду $C_m(k)$ (4), а саме до додавання значень з недіагональних елементів k -го стовпця в $C_s(1)$ до інших елементів у відповідних рядках у $C_m(k)$ (4), тобто значення $n_{i,k}$ і $p_{i,k}$ розподіляються між іншими елементами i -го рядка в $C_s(1)$ з деякими ймовірностями $q_{i,j,k}$. Індeksi i, j, k позначають потенційну залежність ймовірностей $q_{i,j,k}$ від різних класів.

Доведення. З леми 1 сума елементів у кожному рядку в $C_s(1)$ і в $C_m(k)$ (4) повинна бути постійною і дорівнювати 1.

З леми 2, у k -му стовпці порівняння ненульових недіагональних елементів

$n_{i,k}$ та $p_{i,k}$ із C_s (1) та нульових недіагональних елементів $n_{i,k}$ та $p_{i,k}$ із $C_m(k)$ (4), можна отримати, що ненульові недіагональні елементи з C_s (1) фактично розподілені між іншими елементами у відповідних i -х рядках у C_s (1) з деякими ймовірностями $q_{i,j,k}$.

Згідно з лемою 4 це призведе до:

$$p(k)_{i,j} = p_{i,j} + t_{i,k} \times q_{i,j,k}, \quad (6)$$

$$a(k)_{i,j} = a_{i,j} + t_{i,k} \times q_{i,j,k}, \quad (7)$$

$$n(k)_{i,j} = n_{i,j} + t_{i,k} \times q_{i,j,k}, \quad (8)$$

де $t_{i,k}$ означає будь-які недіагональні елементи, такі як $n_{i,k}$ або $p_{i,k}$, k у дужках у $p(k)_{i,j}$, $a(k)_{i,j}$ і $n(k)_{i,j}$ означає, що ці елементи належать до $C_m(k)$ (4), а елементи $p_{i,j}$, $a_{i,j}$, $n_{i,j}$ належать до C_s (1).

Слід зазначити, що

$$\sum_{j=0}^{n-1} q_{i,j,k} = 1, \forall j \neq k. \quad (9)$$

Можна розглянути наступні спрощені “найгірший”, “проміжний” і “найкращий” випадки, які обмежують можливий (найгірший, проміжний і найкращий) вплив на точність за класами:

- “найгірший” розподіл $q_{i,j,k}$ з ймовірністю 0 для агрегування $t_{i,k}$ до діагонального елемента:

$$q_{i,j,k} = 0, \forall i = j, \quad (10a)$$

$$q_{i,j,k} \neq 0, \forall i \neq j; \quad (10b)$$

- “проміжний” розподіл $q_{i,j,k}$ з функцією розподілу ймовірностей f :

$$q_{i,j,k} = f(i, j, k), \forall i, j, k, \quad (11)$$

де слід розглядати декілька часткових випадків функції розподілу ймовірностей, наприклад, рівномірна функція розподілу ймовірностей тощо;

- “найкращий” розподіл $q_{i,j,k}$ з ймовірністю 1 для агрегування $t_{i,k}$ тільки

до діагонального елемента:

$$q_{i,j,k} = 1, \forall i = j, \quad (12a)$$

$$q_{i,j,k} = 0, \forall i \neq j. \quad (12b)$$

Лема 5 (Найгірший вплив мультимодального підходу). У найгіршому випадку застосування мультимодального підходу з “найгіршим” розподілом $q_{i,j,k}$ не підвищує точність за класами.

Доведення. З (7) і (10a) (де $q_{i,j,k} = [i = j] = q_{i,i,k} = 0$) можна отримати точність для кожного i -го класу $a(k)_{i,i}$, отриману за допомогою застосування мультимодального підходу:

$$a(k)_{i,i} = a_{i,i} + t_{i,k} \times q_{i,i,k} = a_{i,i}. \quad (13)$$

Лема 6 (Найкращий вплив мультимодального підходу). У найкращому випадку застосування мультимодального підходу з “найкращим” розподілом $q_{i,j,k}$ підвищує точність за класами на похибку точності $t_{i,k}$.

Доведення. З (7) і (12a) (де $q_{i,j,k} = [i = j] = q_{i,i,k} = 1$) можна отримати точність для кожного i -го класу $a(k)_{i,i}$, отриману за допомогою застосування мультимодального підходу:

$$a(k)_{i,i} = a_{i,i} + t_{i,k} \times q_{i,i,k} = a_{i,i} + t_{i,k}. \quad (14)$$

Лема 7 (Проміжний вплив мультимодального підходу). У найкращому випадку застосування мультимодального підходу з “проміжним” розподілом $q_{i,j,k}$ підвищує точність за класами на похибку точності $t_{i,k}$, помножену на функцію розподілу ймовірностей f .

Доведення. З (7) та (11) (де $q_{i,j,k} = f(i, j, k), \forall i, j, k$) можна отримати точність для кожного i -го класу $a(k)_{i,i}$ отриману шляхом застосування мультимодального підходу:

$$a(k)_{i,i} = a_{i,i} + t_{i,k} \times q_{i,i,k} = a_{i,i} + t_{i,k} \times f(i, j, k). \quad (15)$$

3.4.6. Матриця невідповідностей: Практичні випадки

Для практичних цілей, щоб визначити фактичний вплив мультимодального підходу, можна розглянути деякі часткові випадки, такі як:

- рівномірна функція розподілу ймовірностей для $q_{i,j,k}$ у такому вигляді:

$$q_{i,j,k} = f(i, j, k) = \frac{1}{n-1}, \forall i \neq k; \quad (16)$$

- зміщена рівномірна функція розподілу ймовірностей для $q_{i,j,k}$ зі зміщенням до більшого внеску в істинні результати класифікації, ніж у помилкові, у такому вигляді:

$$q_{i,j,k} = f(i, j, k) = r, \forall i = j, \quad (17a)$$

$$q_{i,j,k} = f(i, j, k) = \frac{1-r}{n-2}, \forall i \neq j, \quad (17b)$$

де $r > \frac{1}{n-1}$ і $r \in [0, 1), \forall i \neq k$.

3.4.6.1. Рівномірна функція розподілу ймовірностей

Теорема 1. Для рівномірної функції розподілу ймовірностей для $q_{i,j,k}$ завдяки застосуванню мультимодального підходу точність багатокласової класифікації для кожного i -го класу $a(k)_{i,i}$ може бути збільшена на помилки класифікації $\frac{t_{i,k}}{n}$.

Доведення. З леми 7, (15) і (16) можна вивести:

$$a_u(k)_{i,i} = a_{i,i} + t_{i,k} \times f(i, i, k) = a_{i,i} + \frac{t_{i,k}}{n-1}, \forall i \neq k. \quad (18)$$

Цей результат можна узагальнити на загальну точність, усереднену для класів, за винятком k -го класу, $A'_s = \frac{1}{n-1} \sum_{i=0}^{n-1} a_{i,i}$, де $i \neq k$, що після застосування мультимодального підходу з рівномірною функцією розподілу ймовірностей можна отримати покращену точність $A_u(k) = \frac{1}{n-1} \sum_{i=0}^{n-1} a(k)_{i,i}$, де $i \neq k$, а нижній індекс u

означає рівномірну функцію розподілу ймовірностей.

Теорема 2. Для рівномірної функції розподілу ймовірностей для $q_{i,j,k}$ завдяки застосуванню мультимодального підходу точність багатокласової класифікації

$$A'_s = \frac{1}{n-1} \sum_{i=0}^{n-1} a_{i,i} \text{ може бути покращена на суму помилок класифікації } \sum_{i=0}^{n-1} t_{i,k} \text{ з}$$

$$\text{остаточним покращенням точності } \Delta A_u(k) = A_u(k) - A'_s = \frac{1}{(n-1)^2} \sum_{i=0}^{n-1} t_{i,k}.$$

Доведення. З леми 7, (15) і (16) можна вивести:

$$\begin{aligned} A_u(k) &= \frac{1}{n-1} \sum_{i=0}^{n-1} a(k)_{i,i} = \\ &= \frac{1}{n-1} \sum_{i=0}^{n-1} \left(a_{i,i} + \frac{t_{i,k}}{n-1} \right) = A'_s + \frac{1}{(n-1)^2} \sum_{i=0}^{n-1} t_{i,k}, \quad (19) \end{aligned}$$

$$\Delta A_u(k) = A_u(k) - A'_s = \frac{1}{(n-1)^2} \sum_{i=0}^{n-1} t_{i,k}, \forall i \neq k, \quad (20)$$

де A'_s – середня точність для всіх класів, крім k -го класу.

3.4.6.2. Зміщена рівномірна функція розподілу ймовірностей

Теорема 3. Для зміщеної рівномірної функції розподілу ймовірностей для $q_{i,j,k}$ завдяки застосуванню мультимодального підходу точність багатокласової класифікації для кожного i -го класу $a(k)_{i,i}$ може бути збільшена на помилки класифікації $\frac{t_{i,k}}{n}$.

Доведення. З леми 7, (15) і (17a) можна вивести:

$$a_{bu}(k)_{i,i} = a_{i,i} + t_{i,k} \times f(i, i, k) = a_{i,i} + t_{i,k} \times r, \forall i \neq k. \quad (21)$$

Аналогічно (як і в теоремі 1) цей результат можна узагальнити на загальну

точність, усереднену для класів, за винятком k -го класу, $A'_s = \frac{1}{n-1} \sum_{i=0}^{n-1} a_{i,i}$, де $i \neq k$,

що після застосування мультимодального підходу зі зміщеною рівномірною

функцією розподілу ймовірностей можна отримати покращену точність $A_{bu}(k) = \frac{1}{n-1} \sum_{i=0}^{n-1} a(k)_{i,i}$, де $i \neq k$, а нижній індекс bu означає зміщену рівномірну функцію розподілу ймовірностей.

Теорема 4. Для зміщеної рівномірної функції розподілу ймовірностей для $q_{i,j,k}$ завдяки застосуванню мультимодального підходу точність багатокласової

класифікації $A'_s = \frac{1}{n-1} \sum_{i=0}^{n-1} a_{i,i}$ може бути покращена на суму помилок класифікації

$\sum_{i=0}^{n-1} t_{i,k}$ з остаточним покращенням точності $\Delta A_{bu}(k) = A_{bu}(k) - A'_s = \frac{r}{n-1} \sum_{i=0}^{n-1} t_{i,k}$.

Доведення. З леми 7, (15) і (17а) можна вивести:

$$\begin{aligned} A_{bu}(k) &= \frac{1}{n-1} \sum_{i=0}^{n-1} a(k)_{i,i} = \\ &= \frac{1}{n-1} \sum_{i=0}^{n-1} (a_{i,i} + t_{i,k} \times r) = A'_s + \frac{r}{n-1} \sum_{i=0}^{n-1} t_{i,k}, \quad (22) \end{aligned}$$

$$\Delta A_{bu}(k) = A_{bu}(k) - A'_s = \frac{r}{n-1} \sum_{i=0}^{n-1} t_{i,k}, \forall i \neq k. \quad (23)$$

Слід зазначити, що для різних значень r можна знайти відповідність з деякими з попередніх результатів:

- якщо $r = 0$, то можна отримати “найгірший” розподіл $q_{i,j,k}$ (лема 5) і з (21) і (23):

$$a_{bu}(k)_{i,i} = a_{i,i} + t_{i,k} \times r = a_{i,i}, \quad (24a)$$

$$\Delta A_{bu}(k) = A_{bu}(k) - A'_s = \frac{r}{n-1} \sum_{i=0}^{n-1} t_{i,k} = 0, \forall i \neq k; \quad (24b)$$

- якщо $r = 1$, то можна отримати “найкращий” розподіл $q_{i,j,k}$ (лема 6) і з (21) і (23):

$$a_{bu}(k)_{i,i} = a_{i,i} + t_{i,k}, \quad (25a)$$

$$\Delta A_{bu}(k) = A_{bu}(k) - A'_s = \frac{1}{n-1} \sum_{i=0}^{n-1} t_{i,k}, \forall i \neq k; \quad (25b)$$

• якщо $r \in (0, 1)$, то можна отримати “проміжний” розподіл $q_{i,j,k}$ (лема 7), а з (21) і (23) можна вважати, що найцікавіший випадок буде пов’язаний із припущенням, що розподіл помилок $t_{i,k}$ з A_s серед усіх класів у $A_{bu}(k)$ має бути

пропорційним середній точності $A_s = \frac{1}{n} \sum_{i=0}^{n-1} a_{i,i}$, тобто $r \sim A_s$:

$$a_{bu}(k)_{i,i} = a_{i,i} + t_{i,k} \times a_{i,i}, \quad (26a)$$

$$\Delta A_{bu}(k) = A_{bu}(k) - A'_s = \frac{A_s}{n-1} \sum_{i=0}^{n-1} t_{i,k}, \forall i \neq k. \quad (26b)$$

3.4.7. Методологія дослідження

Щоб перевірити застосовність часткових випадків, розглянутих у (24) – (26), було проведено навчання з 6-кратним CV для мультимодальних моделей $\forall k \in [0, n]$, де n – кількість класів у наборі даних.

Для перевірки теоретичних співвідношень (18) і (21) для підвищення точності для кожного класу були виміряні та зображені на графіку експериментальні залежності для $\Delta a_u(k)$ і $\Delta a_{bu}(k)$ від $t_{i,k}$:

$$\Delta a_u(k) = a_u(k)_{i,i} - a_{i,i} = \frac{t_{i,k}}{n-1}, \forall i \neq k, \quad (27a)$$

$$\Delta a_{bu}(k) = a_{bu}(k)_{i,i} - a_{i,i} = t_{i,k} \times r, \forall i \neq k. \quad (27b)$$

Таким же чином, щоб перевірити теоретичні співвідношення (20) і (23) для підвищення точності, усередненої за всіма класами, були виміряні та зображені на графіку експериментальні залежності для $\Delta A_u(k)$ і $\Delta A_{bu}(k)$ від $t_{i,k}$:

$$\Delta A_u(k) = A_u(k) - A'_s = \frac{1}{(n-1)^2} \sum_{i=0}^{n-1} t_{i,k}, \forall i \neq k, \quad (28a)$$

$$\Delta A_{bu}(k) = A_{bu}(k) - A'_s = \frac{r}{n-1} \sum_{i=0}^{n-1} t_{i,k}, \forall i \neq k. \quad (28b)$$

3.5. Висновки до розділу 3

Проблема багатокласової класифікації рівня захворювань розглядається для одномодальної (з введенням зображення) моделі та мультимодальної (з введенням зображення та тексту) моделі на основі набору даних RetinaMNIST. Вплив додаткових даних, таких як суб'єктивна думка “пацієнта” про свій стан здоров'я та думка “експерта” (що забезпечує “витік даних” на один із класів), може бути корисним у деяких практичних ситуаціях. Думки пацієнтів та експертів були імітовані додатковими (доповненими) даними, отриманими з змодельованих анкет.

У результаті були створені одномодальна модель (SM) лише з вхідним зображенням і мультимодальні моделі з вхідними зображеннями та текстом, як-от мультимодальна модель із думкою пацієнта (MP), мультимодальна модель із думкою експерта (ME), мультимодальна модель із думкою пацієнта та експерта (MPE) та мультимодальна модель з недетермінованою думкою експерта (MMFE).

На основі створених моделей було проведено дослідження доповнення метаданих для вирішення завдання багатокласової класифікації, дослідження впливу складності мультимодальної моделі на вирішення завдання багатокласової класифікації, дослідження недетермінованого доповнення метаданих для вирішення завдання багатокласової класифікації та дослідження впливу мультимодального доповнення метаданих на точність багатокласової класифікації.

Також було наведено деякі теоретичні оцінки можливих (найкращих, проміжних і найгірших) рівнів покращення точності багатокласової класифікації за допомогою простого математичного опису з акцентом на деяких практичних випадках.

РОЗДІЛ 4

АНАЛІЗ РЕЗУЛЬТАТІВ МЕТОДУ МУЛЬТИМОДАЛЬНОГО АНАЛІЗУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ

4.1. Доповнення метаданих для вирішення завдання багатокласової класифікації

4.1.1. Одномодальна модель (SM)

Модель SM, власне CNN з архітектурою ResNet50 (рис. 3.5.a), була навчена протягом 100 епох у 6-кратному режимі CV. Історії деяких навчальних метрик показані на рис. 4.1, де по горизонтальній осі відкладено кількість епох, а по вертикальній осі – значення точності (рис. 4.1.a) і втрат (рис. 4.1.b). Історії демонструють дуже швидке навчання з меншими стандартними відхиленнями втрат на початковому етапі навчання (до 18 епох). Потім стандартні відхилення втрат стають набагато вищими на пізнішому етапі навчання із зростанням втрат, які можна вважати надмірними (після 50 епох).

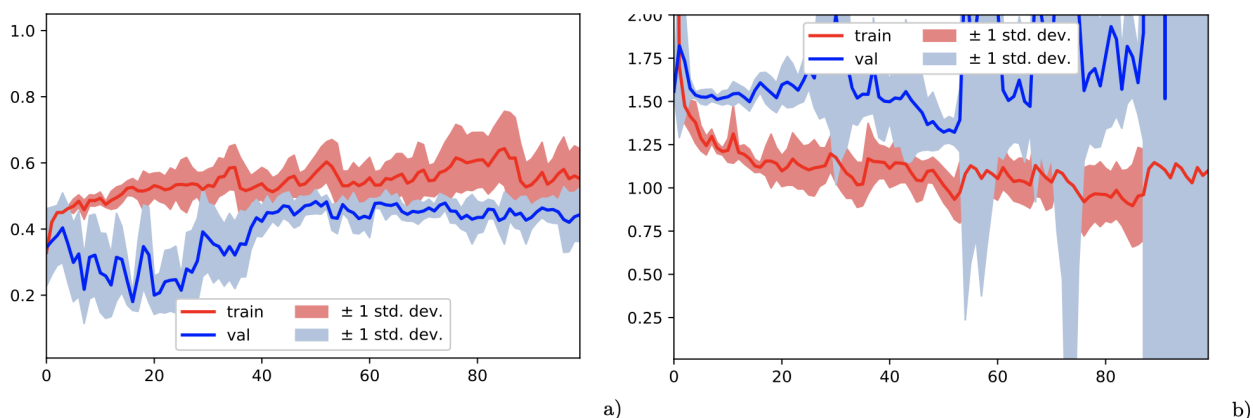


Рисунок 4.1 – Історія точності (a) і втрати (b) після 6-кратної перехресної перевірки для одномодальної моделі

Матриця невідповідностей була розрахована для середнього та стандартного відхилення абсолютних (рис. 4.2.a) і нормалізованих (рис. 4.2.b) значень, отриманих після 6-кратного режиму CV. Також були побудовані криві ROC і розраховані AUC значення із середнім значенням і стандартним відхиленням AUC (рис. 4.3.a).

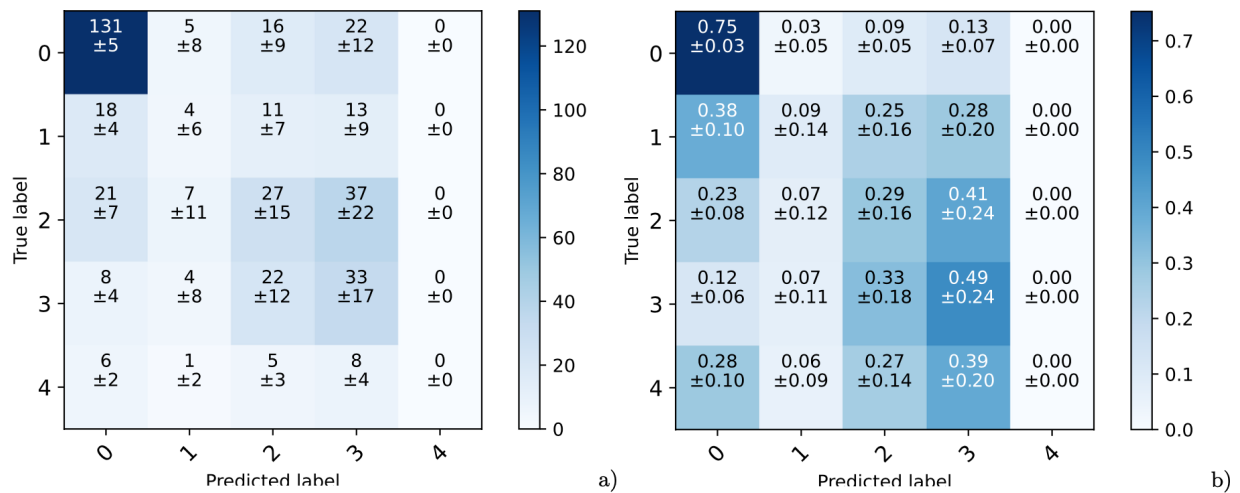


Рисунок 4.2 – Матриця невідповідностей для абсолютних (a) і нормалізованих (b) значень після 6-кратного CV для одномодальної моделі

Також модель SM була навчена протягом 100 епох в режимі OoCV. Крім того, були побудовані криві ROC і розраховані значення AUC із середнім значенням і стандартним відхиленням AUC (рис. 4.3.b).

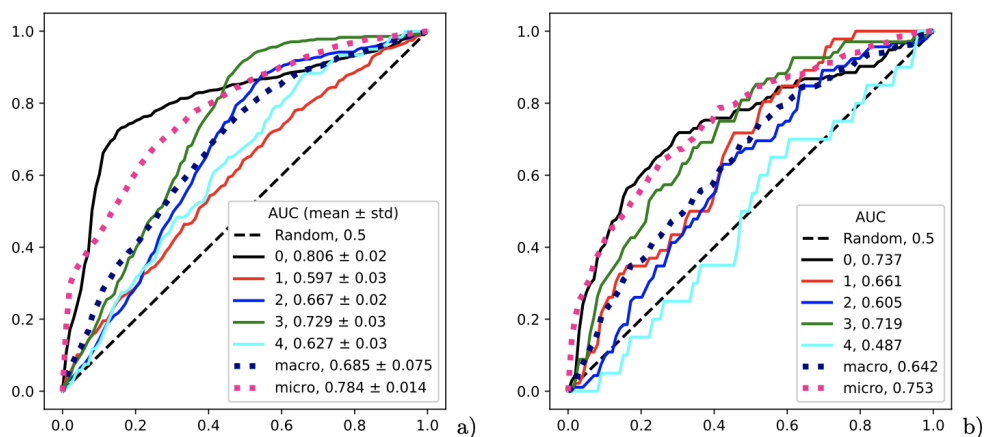


Рисунок 4.3 – Криві ROC із середнім значенням і стандартним відхиленням AUC для моделі SM: після 6-кратного CV (a) і після OoCV (b)

4.1.2. Мультимодальна модель з думкою пацієнта (MP)

Мультимодальна модель з думкою пацієнта (MP), фактично DNN з архітектурами ResNet50 та LSTM (рис. 3.5.b), була навчена протягом 100 епох у процедурах 6-кратного CV та OoCV. Були побудовані криві ROC і розраховані значення AUC із середнім значенням і стандартним відхиленням AUC (рис. 4.4).

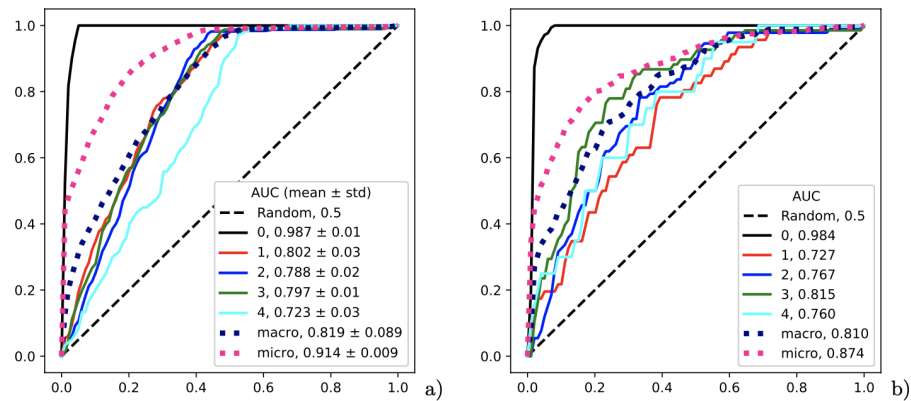


Рисунок 4.4 – Криві ROC із середнім значенням і стандартним відхиленням AUC для моделі MP: після 6-кратного CV (a) і після OoCV (b)

4.1.3. Мультимодальна модель з думкою експерта (ME)

Мультимодальна модель думкою експерта (ME), фактично DNN з архітектурами ResNet50 та LSTM (рис. 3.5.b), була навчена протягом 100 епох у процедурах 6-кратного CV та OoCV. Були побудовані криві ROC і розраховані значення AUC із середнім значенням і стандартним відхиленням AUC (рис. 4.5).

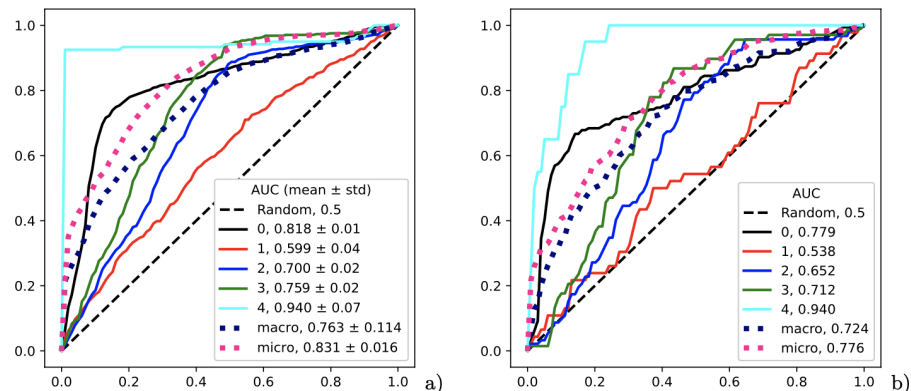


Рисунок 4.5 – Криві ROC із середнім значенням і стандартним відхиленням AUC для моделі ME: після 6-кратного CV (a) і після OoCV (b)

4.1.4. Мультимодальна модель з думкою пацієнта та експерта (MPE)

Мультимодальна модель думкою пацієнта та експерта (MPE), фактично DNN з архітектурами ResNet50 та LSTM (рис. 3.5.b), була навчена протягом 100 епох у процедурах 6-кратного CV та OoCV. Були побудовані криві ROC і розраховані

значення AUC із середнім значенням і стандартним відхиленням AUC (рис. 4.6).

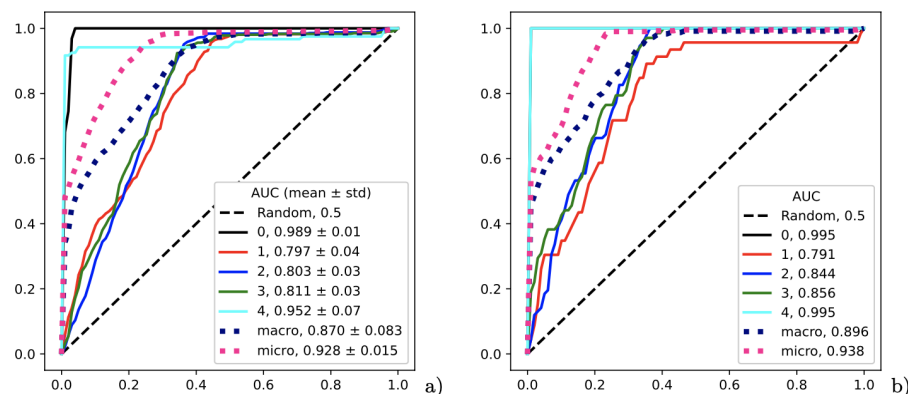


Рисунок 4.6 – Криві ROC із середнім значенням і стандартним відхиленням AUC для моделі MPE: після 6-кратного CV (a) і після OoCV (b)

4.1.5. Аналіз отриманих результатів

У результаті було виявлено, що моделі SM демонструють найнижчі значення, які виділені *курсивом* у табл. 4.1 і 4.2, метрики AUC як для режиму CV (табл. 4.1), так і для режиму OoCV (табл. 4.2). Усі мультимодальні моделі дозволили отримати вищі значення AUC для кожного класу. Найвищі значення AUC виділені **жирним** шрифтом у табл. 4.1 і 4.2. Порожні клітинки в табл. 4.1 і табл. 4.2 означають, що додаткове введення тексту (пацієнтом та/або експертом) дозволяє визначити відповідні класи, і вони недоцільні для обговорення.

Таблиця 4.1 – Середнє значення і стандартне відхилення AUC після 6-кратного CV

Модель	Клас 0	Клас 1	Клас 2	Клас 3	Клас 4	AUC _{macro}	AUC _{micro}
SM	<i>0,806</i> ±0,02	<i>0,597</i> ±0,03	<i>0,667</i> ±0,02	<i>0,729</i> ±0,03	<i>0,627</i> ±0,03	<i>0,685</i> ±0,08	<i>0,784</i> ±0,01
MP	—	0,802 ±0,03	0,788 ±0,02	0,797 ±0,01	0,723 ±0,03	0,819 ±0,09	0,914 ±0,01
ME	0,818 ±0,01	0,599 ±0,04	0,700 ±0,02	0,759 ±0,02	—	0,763 ±0,11	0,831 ±0,02
MPE	—	0,797 ±0,04	0,803 ±0,03	0,811 ±0,03	—	0,870 ±0,08	0,928 ±0,02

Що стосується показників, які спостерігалися в окремих класах після CV,

модель МР дозволила досягти найвищого значення AUC серед усіх моделей для класу 1 (зі збільшенням AUC на 20,5% з 0,597 до 0,802) і класу 4 (зі збільшенням AUC на 9,6% з 0,627 до 0,723). Ці покращення виходять за межі стандартного відхилення в 3% і можуть бути оцінені як значні. Подібним чином модель МЕ дозволила досягти найвищого значення AUC серед усіх моделей для класу 0 (зі збільшенням AUC на 1,2% з 0,806 до 0,818), але це збільшення є близьким до меж стандартного відхилення 1-2% і не може бути оцінене як значне. Модель МРЕ дозволила досягти найвищого значення AUC серед усіх моделей для класу 2 (зі збільшенням AUC на 13,6% з 0,667 до 0,803) і класу 3 (зі збільшенням AUC на 8,2% з 0,729 до 0,811). Ці покращення виходять за межі стандартного відхилення 2-3% і можуть бути оцінені як значні.

Таблиця 4.2 – Значення AUC після OoCV

Модель	Клас 0	Клас 1	Клас 2	Клас 3	Клас 4	AUC _{macro}	AUC _{micro}
SM	0,737	0,661	0,605	0,719	0,487	0,642	0,753
МР	–	0,727	0,767	0,815	0,760	0,810	0,874
МЕ	0,779	0,538	0,652	0,712	–	0,724	0,776
МРЕ	–	0,791	0,844	0,856	–	0,896	0,938

Що стосується показників, які спостерігалися в окремих класах, отриманих після OoCV, модель МР дозволила досягти найвищого значення AUC серед усіх моделей для класу 4 (зі збільшенням AUC на 27,3% з 0,487 до 0,760). Це покращення виходить за межі стандартного відхилення 3% (вимірюного у CV) і може бути оцінене як значне. Модель МЕ дозволила досягти найвищого значення AUC серед усіх моделей для класу 0 (зі збільшенням AUC на 4,2% з 0,737 до 0,779), це покращення виходить за межі стандартного відхилення 1-2% і може бути оцінене як значне. Модель МРЕ дозволила досягти найвищого значення AUC серед усіх моделей для класу 1 (зі збільшенням AUC на 13% з 0,661 до 0,791), класу 2 (зі збільшенням AUC на 23,9% з 0,605 до 0,844), і класу 3 (зі збільшенням AUC на 13,7% з 0,719 до 0,856). Ці покращення виходять за межі стандартного відхилення 2-3% і можуть бути оцінені як значні.

Подібні покращення значень AUC_{macro} і AUC_{micro} не можна оцінити та обговорити однаково, оскільки вони сильно залежать від класів, явно “мета-промаркованих” додатковим текстовим введенням від пацієнта та/або експерта (які позначаються як порожні клітинки в табл. 4.1 і 4.2) для відповідних моделей. Таке явне “мета-маркування” забезпечує витік даних, що дозволяє точно передбачити такі класи та значно покращити значення AUC_{macro} і AUC_{micro} .

Порівняння абсолютно найкращих значень AUC, які виділені **червоним** шрифтом у табл. 4.1 і 4.2, для CV (табл. 4.1) і OoCV (табл. 4.2) може створити викривлене враження про вищий приріст точності після OoCV. Проблема полягає в тому, що вибір найкращих моделей у режимі OoCV проводився після однієї спроби перевірки (на 120 зображеннях) у порівнянні з вибором найкращих моделей у режимі CV, виконаного після 6 спроб перевірки (на 180 зображеннях) та їх усереднення. Це означає, що результати з табл. 4.1 є статистично надійними в межах стандартних відхилень, але результати з табл. 4.2 є лише прикладами деяких можливих коливань показників (тут AUC).

4.2. Вплив складності мультимодальної моделі на вирішення завдання багатокласової класифікації

4.2.1. Одномодальна модель (SM)

Різні версії моделі SM, власне різні види архітектур CNN (табл. 3.2), були навчені протягом 100 епох у режимі 6-кратного CV. Параметр конфігурації для k-кратної перехресної перевірки, k, визначає кількість частин, на які було розділено набір даних. Найпопулярнішим значенням, яке використовується в цілях машинного навчання, є $k = 10$, але тут $k = 6$, щоб забезпечити більше зразків для частин для перевірки. Значення AUC для кожного класу (табл. 4.5) розраховувалось з визначенням їх середнього значення та стандартного відхилення. Також значення AUC_{macro} і AUC_{micro} були визначені (табл. 4.4) і візуалізовані у вигляді стовпчастої діаграми (рис. 4.7).

Таблиця 4.3 – Значення AUC після 6-кратного CV та OoCV для моделі SM

CNN	6-кратне CV		OoCV	
	AUC _{macro}	AUC _{micro}	AUC _{macro}	AUC _{micro}
ResNet50	0,697±0,074	0,796±0,006	0,667	0,759
MobileNetV2	0,503±0,007	0,707±0,015	0,500	0,708
NASNetLarge	0,556±0,027	0,688±0,043	0,500	0,647
EfficientNetB7	0,658±0,092	0,757±0,009	0,680	0,782
DenseNet201	0,704±0,073	0,798±0,010	0,722	0,794
VGG19	0,610±0,069	0,745±0,024	0,499	0,721
EfficientNetV2L	0,613±0,068	0,738±0,023	0,659	0,776
NASNetMobile	0,619±0,074	0,738±0,021	0,617	0,744
EfficientNetB0	0,693±0,079	0,781±0,022	0,669	0,752
EfficientNetV2S	0,647±0,095	0,765±0,007	0,690	0,767

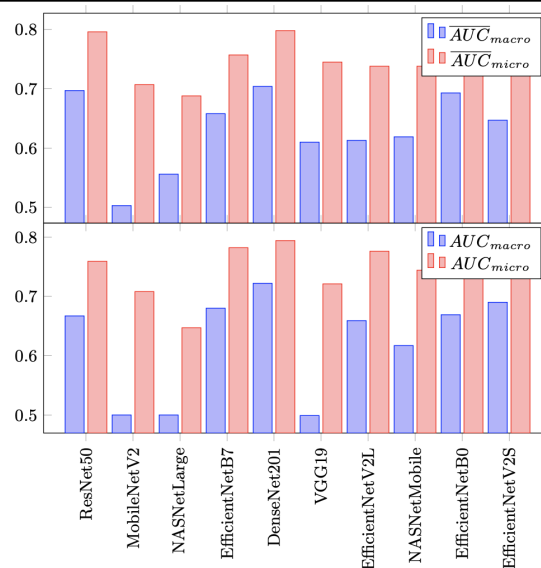


Рисунок 4.7 – Значення AUC для моделі SM після 6-кратного CV (внизу) та OoCV (вгорі)

Також ці версії моделі SM з різними видами архітектури CNN (табл. 3.2) були навчені протягом 100 епох у режимі OoCV. Потім були визначені значення AUC для кожного класу (табл. 4.5), а також значення AUC_{macro} і AUC_{micro} (табл. 4.3), які були візуалізовані у вигляді стовпчастої діаграми (рис. 4.7).

4.2.2. Мультимодальна модель з думкою пацієнта (MP)

Подібним чином, як і з моделями SM в розділі 4.2.1, різні версії мультимодальної моделі з думкою пацієнта (MP), фактично різні типи архітектури згорткової нейронної мережі і рекурентної нейронної мережі з архітектурою LSTM (табл. 3.2), тренувалися протягом 100 епох у режимі 6-кратного CV і в режимі OoCV. Потім були визначені значення AUC для кожного класу (табл. 4.5), а також значення AUC_{macro} і AUC_{micro} (табл. 4.4), які були візуалізовані у вигляді стовпчастої діаграми (рис. 4.8).

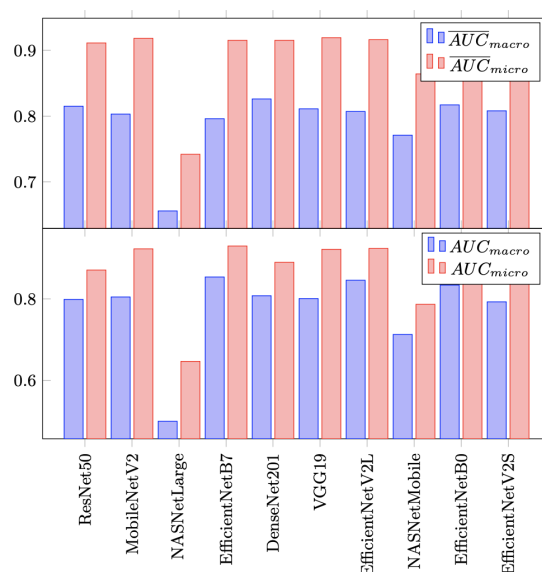


Рисунок 4.8 – Значення AUC для моделі MP після 6-кратного CV (внизу) та OoCV (вгорі)

Таблиця 4.4 – Значення AUC після 6-кратного CV та OoCV для моделі MP

CNN	6-кратне CV		OoCV	
	AUC_{macro}	AUC_{micro}	AUC_{macro}	AUC_{micro}
ResNet50	0,815±0,089	0,911±0,007	0,799	0,871
MobileNetV2	0,803±0,098	0,918±0,004	0,805	0,923
NASNetLarge	0,656±0,082	0,742±0,088	0,500	0,647
EfficientNetB7	0,796±0,103	0,915±0,009	0,854	0,930
DenseNet201	0,826±0,085	0,915±0,006	0,808	0,890

Продовження табл. 4.4

VGG19	0,811±0,093	0,919±0,005	0,801	0,922
EfficientNetV2L	0,807±0,096	0,916±0,003	0,846	0,924
NASNetMobile	0,771±0,119	0,864±0,036	0,713	0,787
EfficientNetB0	0,817±0,094	0,922±0,006	0,834	0,930
EfficientNetV2S	0,808±0,100	0,917±0,008	0,793	0,921

4.2.3. Аналіз отриманих результатів

Усі моделі МР з різними видами архітектури CNN (за винятком NASNetLarge) дозволили досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC для класу 1 у діапазоні від 15% до 26%, що виходить за межі стандартного відхилення 3-4%, вимірюного за допомогою перехресної перевірки, і може бути оцінене як значуще. Для класу 2 усі моделі МР продемонстрували значні покращення від 11% до 28%, які також виходять за межі стандартного відхилення 3-4%. Що стосується класів 3 і 4, лише MobileNetV2 і EfficientNetV2L демонструють значні покращення від 22% до 26% (MobileNetV2) і від 12% до 14% (EfficientNetV2L), які також виходять за межі стандартного відхилення 3% (MobileNetV2) і 8% (EfficientNetV2L).

4.2.3.1. Значення AUC_{macro} в порівнянні зі складністю моделі

У режимі CV було виявлено, що кілька архітектур DNN дозволили досягти найвищих значень AUC_{macro} (рис. 4.9) серед усіх архітектур після 6-кратного CV для моделі SM. Найвище значення AUC знаходиться в діапазоні 0,693-0,704 для ResNet50, DenseNet201 і EfficientNetB0. Що стосується точності, що спостерігається для моделі МР, було виявлено, що ті самі архітектури, що й для моделі SM, дозволили досягти найвищого значення AUC серед усіх архітектур після 6-кратного CV. Найвище значення AUC знаходиться в діапазоні 0,815-0,826

для ResNet50, DenseNet201 і EfficientNetB0. Порівнюючи розмір архітектур з найкращим значенням AUC, можна виявити, що дві з них мають середній розмір (ResNet50 і DenseNet201) і одна – малий розмір (EfficientNetB0). На рис. 4.9 зображені середні значення AUC_{macro} із стандартними відхиленнями (показаними у вигляді смужок помилок) у порівнянні з кількістю параметрів моделі N (у тисячах), показані на логарифмічній шкалі як $\log_2(N)$.

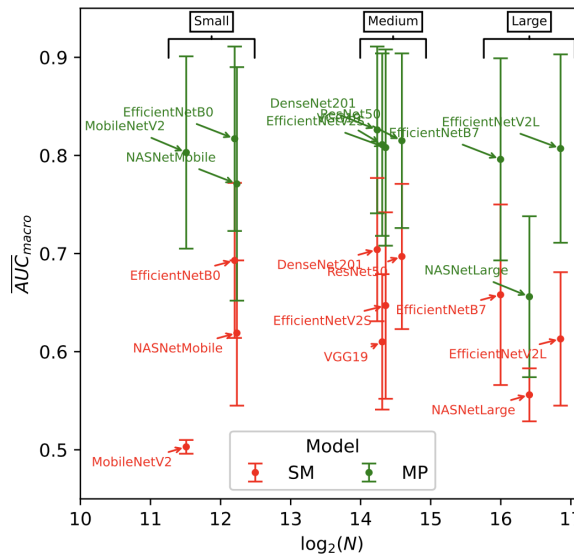


Рисунок 4.9 – Середні значення AUC_{macro} із стандартними відхиленнями у порівнянні з кількістю параметрів моделі N

У режимі OoCV для моделі SM було виявлено, що лише архітектура DenseNet201 дозволяє досягти найвищого значення AUC. Це значення становить 0,722, друге найвище значення – лише 0,690. Однак є дві найкращі архітектури для моделі MP. Це EfficientNetB7 та EfficientNetV2L зі значеннями AUC, що дорівнюють 0,854 та 0,846 відповідно. Порівнюючи розмір архітектур з найкращим значенням AUC, можна виявити, що дві з них мають великий розмір (EfficientNetB7 і EfficientNetV2L) і одна – середній розмір (DenseNet201).

Загалом серед усіх моделей DNN три групи їхніх компонентів CNN можна класифікувати за розміром (рис. 4.9):

- малі: MobileNetV2, EfficientNetB0, NASNet-Mobile;
- середні: VGG19, ResNet50, DenseNet201, EfficientNetV2S;
- великі: EfficientNetB7, EfficientNetV2L, NASNetLarge.

З табл. 4.3, табл. 4.4 та рис. 4.9 видно, що застосування додаткової

модальності в моделях МР дозволяє отримати вищу точність щодо значень \overline{AUC}_{macro} . Насправді значення \overline{AUC}_{macro} для моделей МР набагато більші, ніж значення \overline{AUC}_{macro} для моделей SM порівняно зі спостережуваними стандартними відхиленнями. Цікаво, що приріст точності може бути досить різним для різних моделей DNN, наприклад, для NASNetLarge приріст точності не дуже виражений, а для MobileNetV2 він найбільший серед усіх.

4.2.3.2. Значення AUC для класів в порівнянні зі складністю моделі

Завдяки певній візуалізації (рис. 4.10) для значень AUC для кожного класу (табл. 4.5) можна спостерігати деякі відмінності покращення точності $\Delta = \overline{AUC}^{MP} - \overline{AUC}^{SM}$ серед категорій моделей за розміром для різних класів. Наприклад, ієрархію підвищення точності Δ можна спостерігати так: $\Delta(\text{Клас 1})$ (червоний на рис. 4.10) $> \Delta(\text{Клас 2})$ (зелений на рис. 4.10) $> \Delta(\text{Клас 3})$ (синій на рис. 4.10) $\geq \Delta(\text{Клас 4})$ (чорний на рис. 4.10). У цій ієрархії всі маленькі моделі для класів 1 і 2 демонструють найкращі покращення точності Δ , які є набагато вищими за стандартні відхилення (“std” на рис. 4.10) як усереднені (“aver”) або максимальні (“max”) значення, оцінені за всіма значеннями стандартного відхилення для всіх моделей з табл. 4.5. Крім того, малі та середні моделі для класів 3 і 4 демонструють лише трохи краще покращення точності порівняно з великими моделями. Це означає, що додавання “витоку даних” для класу 0 у моделях МР дозволило отримати більш високе покращення точності Δ для найближчих класів 1 і 2 (які подібні до класу 0), але Δ було не таким високим для далеких класів 3 і 4 (які більше відрізняються від класу 0).

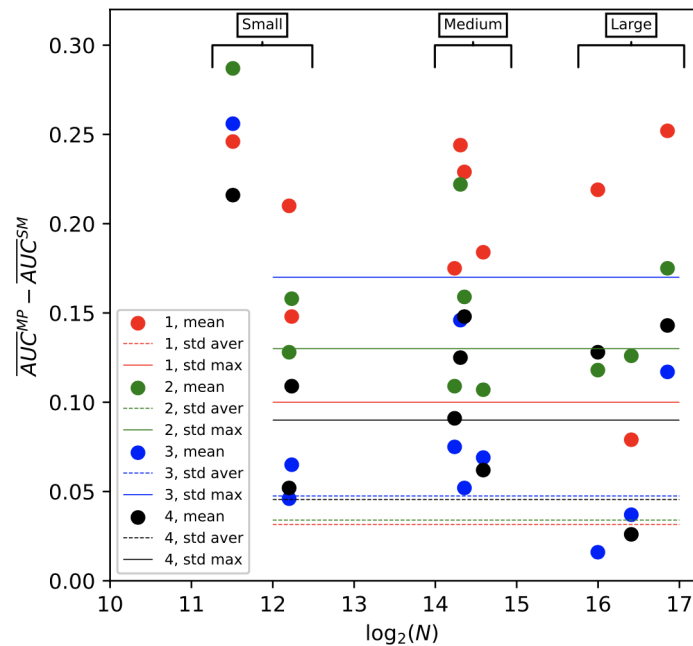


Рисунок 4.10 – Підвищення точності багатокласової класифікації як різниця середніх значень AUC у порівнянні з кількістю параметрів моделі N

На рис. 4.10 кількість параметрів моделі N (у тисячах), показана на логарифмічній шкалі як $\log_2(N)$ для кожного класу, згаданого в легенді. Легенда містить номер класу, тип значення (“mean” або “std”), де “std” може бути середнім (“aver”) або максимальним значенням (“max”) за всіма значеннями стандартного відхилення з табл. 4.5.

Таблиця 4.5 – Значення AUC після 6-кратного CV

CNN	Модель	Клас 0	Клас 1	Клас 2	Клас 3	Клас 4
ResNet50	SM	0,818±0,01	0,600±0,04	0,668±0,01	0,734±0,02	0,663±0,04
	MP	0,986±0,01	0,784±0,04	0,775±0,02	0,803±0,01	0,725±0,06
MobileNetV2	SM	0,502±0,00	0,499±0,00	0,496±0,01	0,502±0,02	0,517±0,03
	MP	0,995±0,00	0,745±0,01	0,783±0,00	0,758±0,01	0,733±0,02
NASNetLarge	SM	0,572±0,10	0,505±0,02	0,551±0,05	0,581±0,09	0,570±0,08
	MP	0,807±0,22	0,584±0,10	0,677±0,13	0,618±0,12	0,596±0,07
EfficientNetB7	SM	0,783±0,03	0,536±0,02	0,671±0,02	0,725±0,03	0,574±0,03
	MP	0,995±0,00	0,755±0,04	0,789±0,01	0,741±0,04	0,702±0,03

Продовження табл. 4.5

DenseNet201	SM	0,825±0,01	0,625±0,02	0,676±0,01	0,745±0,02	0,647±0,03
	MP	0,989±0,01	0,800±0,02	0,785±0,02	0,820±0,03	0,738±0,03
VGG19	SM	0,726±0,12	0,520±0,03	0,565±0,12	0,617±0,17	0,621±0,07
	MP	0,995±0,00	0,764±0,03	0,787±0,01	0,763±0,03	0,746±0,04
EfficientNetV2L	SM	0,715±0,03	0,513±0,02	0,608±0,06	0,654±0,08	0,578±0,06
	MP	0,995±0,00	0,765±0,03	0,783±0,01	0,771±0,02	0,721±0,02
NASNetMobile	SM	0,715±0,13	0,522±0,03	0,607±0,07	0,692±0,10	0,560±0,09
	MP	0,995±0,00	0,670±0,07	0,765±0,02	0,757±0,04	0,669±0,04
EfficientNetB0	SM	0,811±0,01	0,581±0,03	0,663±0,05	0,747±0,02	0,662±0,02
	MP	0,995±0,00	0,791±0,02	0,791±0,01	0,793±0,04	0,714±0,04
EfficientNetV2S	SM	0,779±0,03	0,552±0,04	0,624±0,04	0,735±0,03	0,545±0,07
	MP	0,995±0,00	0,781±0,02	0,783±0,01	0,787±0,03	0,693±0,04

4.3. Недетерміноване доповнення метаданих для вирішення завдання багатокласової класифікації

Для моделі F_N (“одна детермінована думка експерта”) середнє значення $\overline{A_{m,s}}$ і стандартне відхилення $\sigma(A_{m,s})$ AUC були розраховані після експериментів CV для різних класів (N) для наборів даних RetinaMNIST (рис. 4.11.a), DermaMNIST (рис. 4.11.b), BloodMNIST (рис. 4.11.c), PathMNIST (рис. 4.11.d) і CIFAR10 (рис. 4.11.e). Нижній індекс s позначає одномодальний експеримент (SINGLE), а нижній індекс m – різні мультимодальні експерименти (F_N). “std” в легенді (рис. 4.11) означає $\sigma(A_s)$ лише для експерименту SINGLE. Можна побачити, що моделі SINGLE і F_N, які використовуються для різних наборів даних, призводять до різних рівнів точності для різних класів і наборів даних.

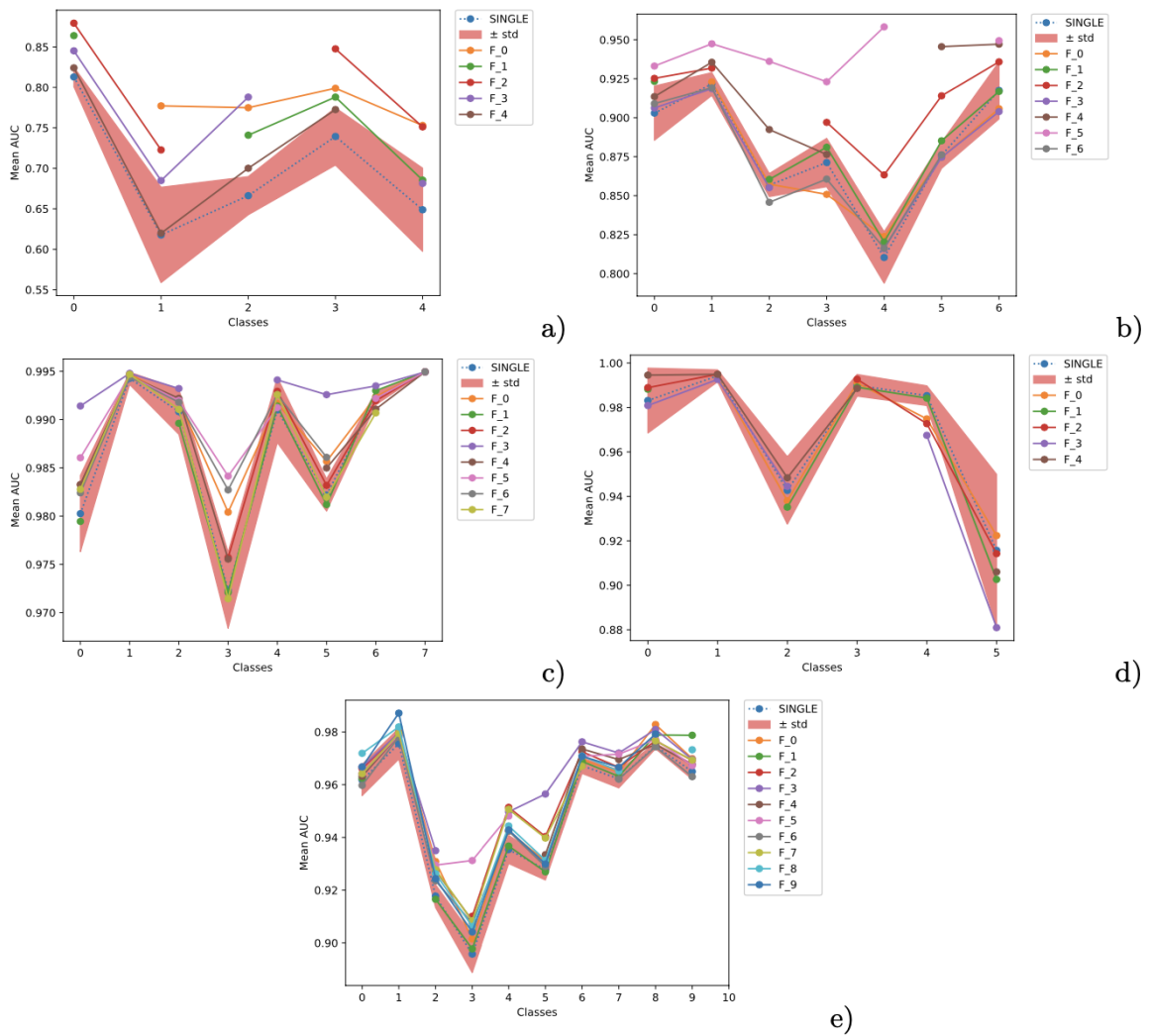


Рисунок 4.11 – Середні значення AUC після CV для одномодального експерименту (SINGLE) та різних мультимодальних експериментів (F_N) для різних класів (N) для різних наборів даних

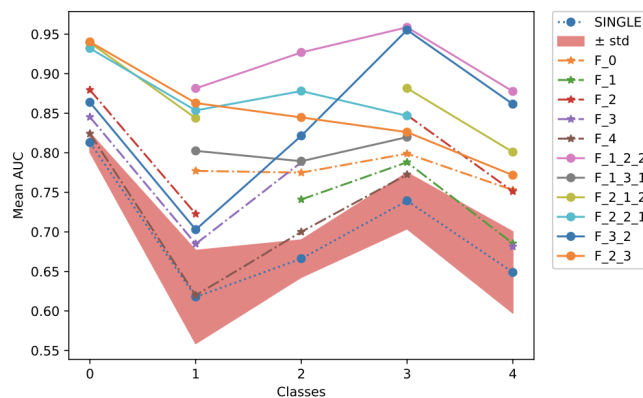


Рисунок 4.12 – Середні значення AUC після CV для одномодального (SINGLE) і мультимодальних експериментів (F_N, F_k_l і F_k_l_m) для набору даних RetinaMNIST

Подібним чином для шаблонів F_k_l або $F_k_l_m$ (“декілька недетермінованих думок експерта”) значення \overline{A}_m і $\sigma(A_m)$ були розраховані після CV (рис. 4.12, табл. 4.6).

Таблиця 4.6 – Значення AUC після CV для набору даних RetinaMNIST

Модель	Клас 0	Клас 1	Клас 2	Клас 3	Клас 4	AUC _{macro}	AUC _{micro}
Single	0,81±0,01	0,62±0,06	0,67±0,02	0,74±0,03	0,65±0,03	0,70±0,07	0,79±0,02
F_0	–	0,78±0,02	0,78±0,02	0,80±0,04	0,75±0,04	0,82±0,09	0,91±0,01
F_1	0,86±0,02	–	0,74±0,02	0,79±0,03	0,69±0,06	0,81±0,10	0,89±0,01
F_2	0,88±0,01	0,72±0,02	–	0,85±0,02	0,75±0,01	0,84±0,09	0,91±0,01
F_3	0,85±0,02	0,69±0,02	0,79±0,02	–	0,68±0,06	0,80±0,11	0,89±0,01
F_4	0,82±0,02	0,62±0,03	0,70±0,01	0,77±0,03	–	0,77±0,11	0,83±0,01
F_1_2_2	–	0,88±0,02	0,93±0,03	0,96±0,02	0,88±0,05	0,93±0,05	0,97±0,01
F_1_3_1	–	0,80±0,02	0,79±0,02	0,82±0,02	–	0,85±0,07	0,92±0,01
F_2_1_2	0,94±0,01	0,84±0,03	–	0,88±0,05	0,80±0,06	0,88±0,05	0,93±0,02
F_2_2_1	0,93±0,01	0,85±0,02	0,88±0,01	0,85±0,02	–	0,89±0,04	0,93±0,01
F_2_3	0,94±0,01	0,86±0,03	0,85±0,01	0,83±0,02	0,77±0,04	0,85±0,06	0,92±0,01
F_3_2	0,86±0,01	0,70±0,03	0,82±0,01	0,95±0,01	0,86±0,04	0,84±0,08	0,90±0,02

Аналогічно, для окремої моделі F_super20, яка використовується лише для набору даних CIFAR100, значення $\overline{A}_{m,s}$ і $\sigma(A_{m,s})$ були розраховані після CV (рис. 4.13, табл. 4.7).

Кольори “std” в легенді (рис. 4.13) позначають межі стандартних відхилень для відповідних експериментів: SINGLE, $\sigma(A_s)$ (синій) і F_super20, $\sigma(A_m)$ (червоний).

У результаті дослідження було встановлено, що всі мультимодальні моделі з FMDA (F_N) дозволили отримати вищі значення AUC для кожного класу в порівнянні з одномодальною моделлю (SINGLE) (рис. 4.11).

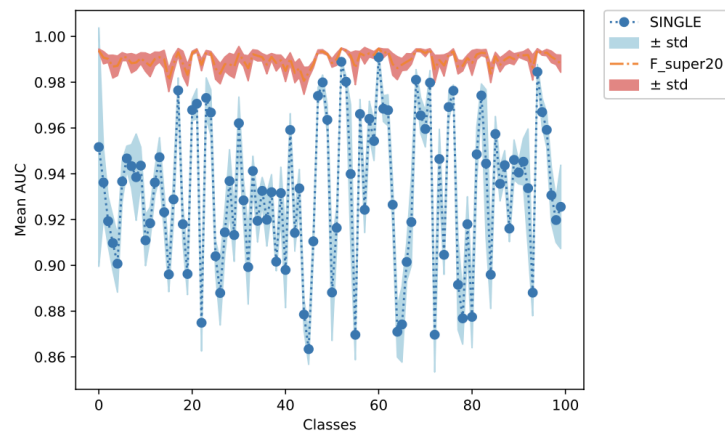


Рисунок 4.13 – Середні значення AUC для різних класів після CV для одномодального (SINGLE) та мультимодального (F_20) експерименту для набору даних CIFAR100

Таблиця 4.7 – Значення AUC після CV для набору даних CIFAR100

Модель	AUC _{macro}	AUC _{micro}
Single	0,932±0,031	0,935±0,010
F_20	0,990±0,003	0,992±0,001

Різниця (приріст точності) $\Delta A(N) = \overline{A_m(N)} - \overline{A_s(N)}$ між середніми значеннями AUC для експерименту SINGLE ($\overline{A_s(N)}$) і середніми значеннями AUC для різних F_N мультимодальних експериментів ($\overline{A_m(N)}$) були розраховані та нанесені на графік (рис. 4.14-4.15). для різних класів (N) для наборів даних RetinaMNIST (рис. 4.14.a), DermaMNIST (рис. 4.14.b), BloodMNIST (рис. 4.15.a), PathMNIST (рис. 4.15.b) і CIFAR10 (рис. 4.15.c). Рядки в легенді рис. 4.14-4.15 означають: $\sigma_{\max}(A_{\text{macro}})$, $\sigma_{\max}(A_{\text{micro}})$, $\sigma_{\max}(A_s(N))$ – максимальні значення стандартного відхилення для значень A_{macro} , A_{micro} , $A_s(N)$ (відповідно) серед усіх експериментів. Пунктирна лінія “No changes” наведена як орієнтир для очей.

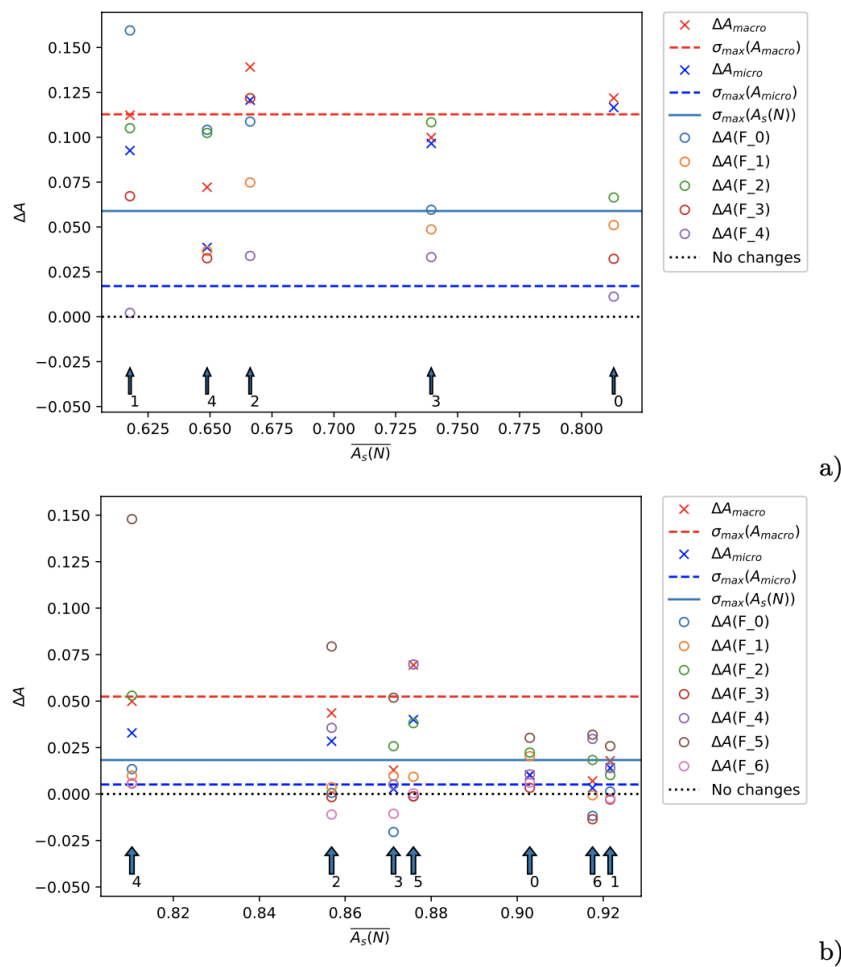


Рисунок 4.14 – $\Delta A(N)$ для різних мультимодальних експериментів (F_N) для різних класів (N) для наборів даних RetinaMNIST (a) і DermaMNIST (b)

Слід зазначити, що для кожного набору даних кращий приріст точності $\Delta A(N)$ спостерігався для важче розпізнаваних класів у кожному наборі даних. Усі мультимодальні моделі з FMDA демонструють кращий приріст точності для наступних конкретних наборів медичних даних у порядку $\Delta A(N)$: RetinaMNIST, DermaMNIST, PathMNIST і BloodMNIST. Цей порядок корелює з порядком $A_s(N)$ для цих наборів даних. Це означає, що FMDA дозволяє отримати кращий приріст точності для важче розпізнаваних наборів даних, що дуже важливо для практики. Ось чому подальше обговорення буде зосереджено з більш детальною інформацією щодо набору даних RetinaMNIST (з CIFAR100 для порівняння), який демонструє найнижчу точність за $A_s(N)$ для одномодальної моделі (SINGLE) і найвищий приріст точності за $\Delta A(N)$ для мультимодальних моделей із FMDA.

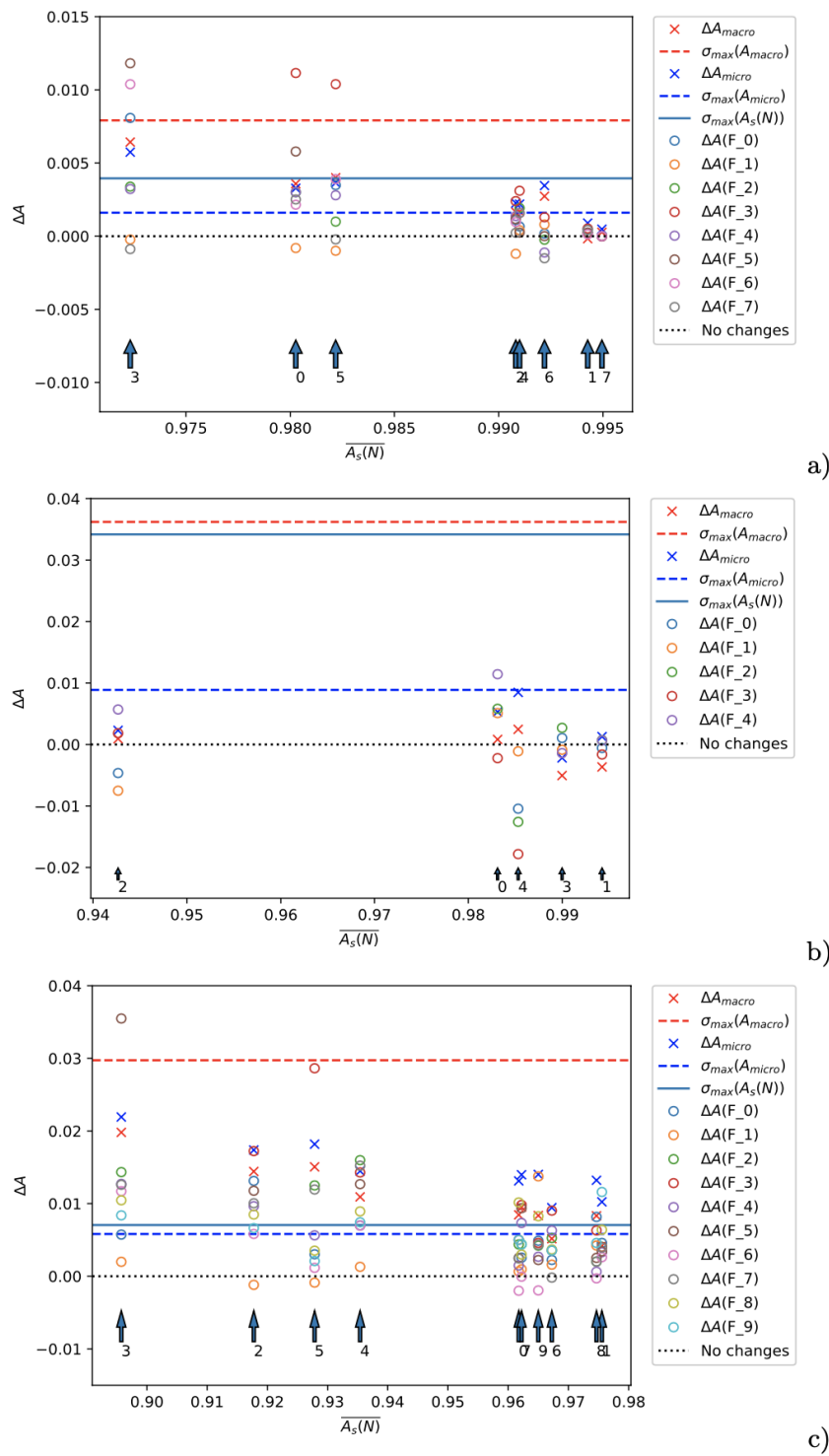


Рисунок 4.15 – $\Delta A(N)$ для різних мультимодальних експериментів (F_N) для різних класів (N) для наборів даних BloodMNIST (a), PathMNIST (b), CIFAR10 (c)

Найвищі значення AUC виділені жирним шрифтом у табл. 4.6-4.7. Порожні клітинки в табл. 4.6 означають, що додаткове текстове маркування дозволяє точно визначити відповідні класи (прямий “витік даних”), і вони марні для обговорення. Наприклад, для окремих класів у наборі даних RetinaMNIST наступний найвищий приріст точності ΔA спостерігався для моделі F_1_2_2 із FMDA (табл. 4.6, рис. 4.12,

рядок F_1_2_2): 26% для класу 1 (від 0,62 до 0,88) і для класу 2 (від 0,67 до 0,93), 22% для класу 3 (від 0,74 до 0,96) і 23% для класу 4 (від 0,65 до 0,88). Що стосується класу 0, кілька мультимодальних моделей F_2_1_2, F_2_2_1 і F_2_3 продемонстрували аналогічний приріст точності на 12-13% (від 0,81 до 0,93-0,94) (табл. 4.6, рис. 4.12, рядки F_2_1_2, F_2_2_1 і F_2_3). Це можна пояснити “витоком даних” маркуванням FMDA всередині пар-сімейств близьких і важковпізнаваних класів (0 і 1, 1 і 2, 2 і 3, 3 і 4). Це стимулює DNN до пошуку прихованих “спільних” ознак серед класів у сімействах і, в той же час, до пошуку прихованих “відмінних” ознак, що “відрізняють” ці сімейства одне від одного. Це явище можна продемонструвати за допомогою моделі F_super20, яка використовується з маркуванням FMDA 20 сімействами-суперкласами з набором даних CIFAR100 (табл. 4.7, рис. 4.13).

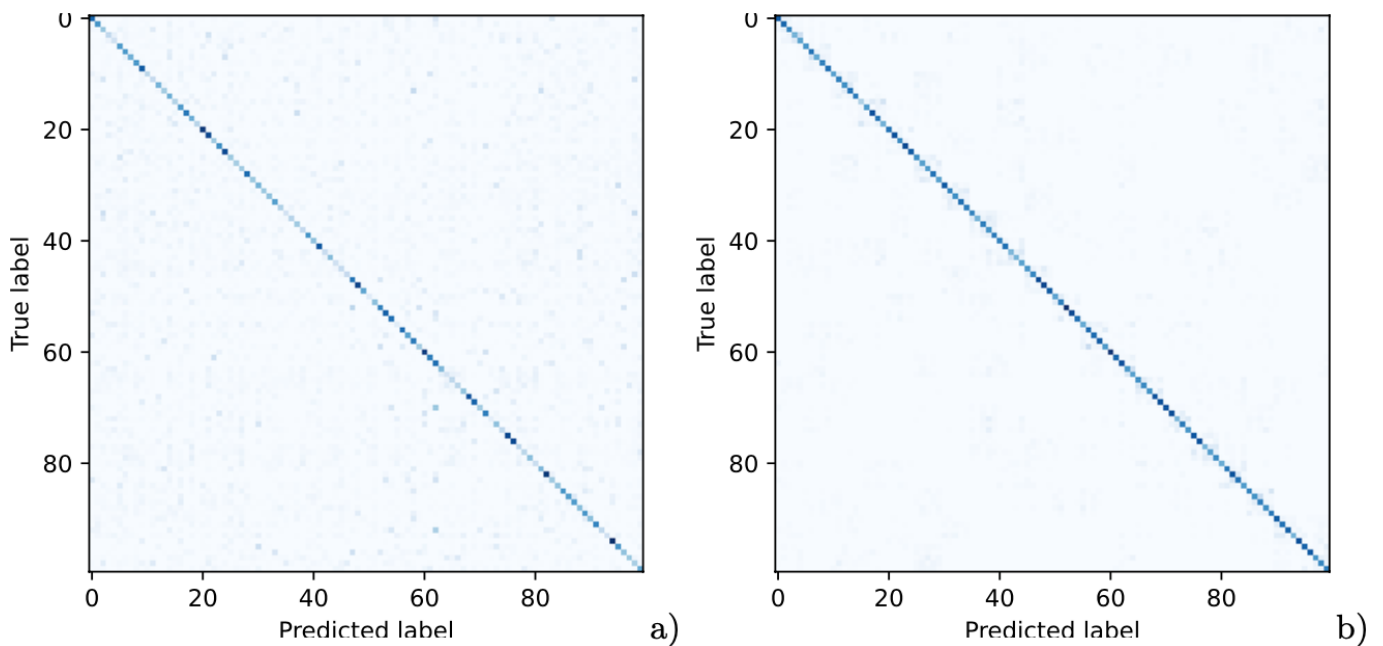


Рисунок 4.16 – Матриця невідповідностей для CIFAR100 для експериментів з однією модальністю (a) і мультимодальністю (F_20) (b) для різних класів

Приріст точності $\Delta A(N)$ спостерігався для всіх 100 класів від 0,1 до 11% (рис. 4.13), ΔA_{macro} до 5,8%, а ΔA_{micro} до 5,7%. Крім того, рівень невизначеності через мінливість даних, виміряних стандартним відхиленням, зменшився приблизно в 10 разів для $\sigma(A_{macro})$ з 0,031 до 0,003, а для $\sigma(A_{micro})$ з 0,010 до 0,001 (табл. 4.7). Знову ж таки, це можна пояснити вищезгаданим “витоком даних” за

допомогою маркування FMDA між і всередині сімейств-суперкласів 5 близьких і важко впізнаваних класів всередині кожного суперкласу. Кількість хибно позитивних і хибно негативних прогнозів зменшується, особливо між 20 сімействами-суперкласами, але кількість істинно позитивних прогнозів зростає, що можна побачити на матрицях невідповідностей (рис. 4.16).

Загалом, більшість покращень, отриманих за допомогою FMDA, виходять за межі стандартних відхилень і можуть бути оцінені як значні. У практичному сенсі це означає, наприклад, для набору даних RetinaMNIST і проблеми, що медичний експерт може отримати потужний інструмент для розрізнення кількох подібних класів від деякого сімейства-суперкласу класів, якщо експерт впевнений, що досліджуване зображення відноситься до цього сімейства-суперкласу і має модель, навчену маркуванням FMDA для такого сімейства-суперкласу. Для деяких класів у найскладнішому наборі даних RetinaMNIST від MedMNIST принаймні FMDA дає приріст точності від 12 до 26%. FMDA можна ефективно використовувати на додаток до попередніх спроб підвищити точність моделей багатокласової класифікації за допомогою різних методів доповнення даних, оптимізації розміру партії, налаштування гіперпараметрів, оптимізації розміру зображення, розміру набору даних у медичних і більш загальних програмах.

4.4. Вплив мультимодального доповнення метаданих на точність багатокласової класифікації

Залежності, описані в розділі 3.4.6, були виміряні та зображені на графіках для моделей, навчених на кількох наборах даних PathMNIST (рис. 4.17), RetinaMNIST (рис. 4.18) і CIFAR10 (рис. 4.19).

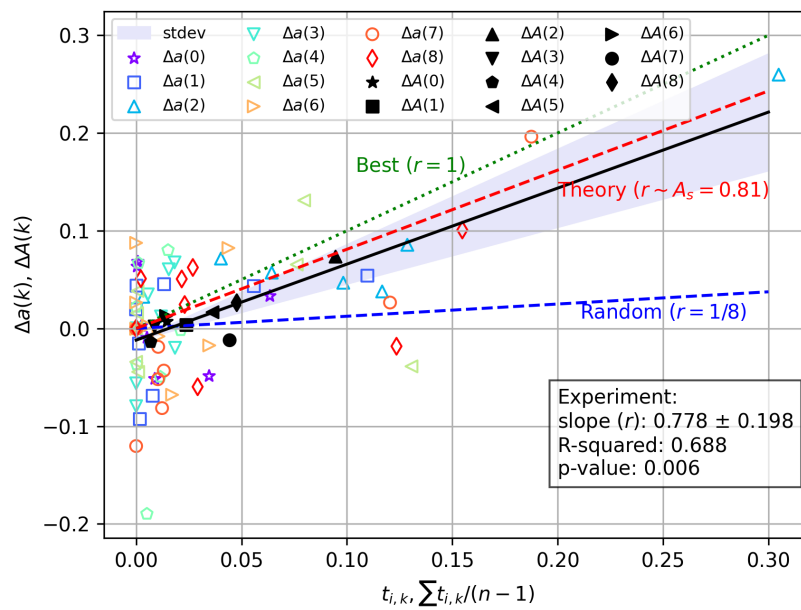


Рисунок 4.17 – Залежність $\Delta a(k)$ та $\Delta A(k)$ від $t_{i,k}$ та $\sum_{k=1}^{n-1} t_{i,k}$ для набору даних

PathMNIST

На рис. 4.17–4.19 випадковий сценарій для $r = 0$ (24) позначено пунктирною синьою лінією, і його можна досягти у випадку, коли похибка точності $t_{i,k}$ розподіляється серед усіх інших членів $a_{i,j}$ і фактично демонструє найгірший практично можливий сценарій підвищення точності. Абсолютно найкращий сценарій для $r = 1$ (25) позначено пунктирною зеленою лінією, але його можна досягти лише у випадку, коли похибка точності $t_{i,k}$ переходить лише до діагонального члена $a_{i,i}$, що навряд чи можливо на практиці. Проміжний сценарій для $r = \frac{1}{n}$ (26) позначено пунктирною червоною лінією, але він можливий у випадку, коли похибка точності $t_{i,k}$ розподіляється пропорційно A_s серед усіх інших членів $a_{i,j}$.

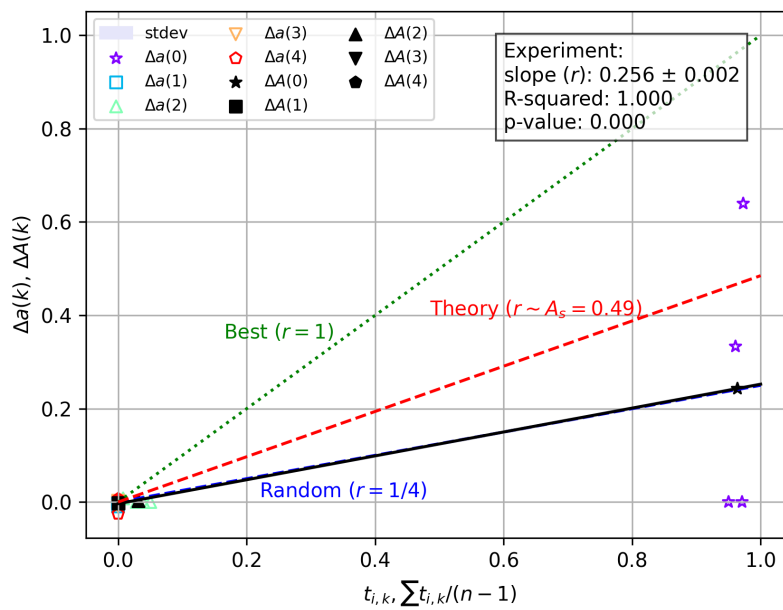


Рисунок 4.18 – Залежність $\Delta a(k)$ та $\Delta A(k)$ від $t_{i,k}$ та $\sum_{i=0}^{n-1} \frac{t_{i,k}}{n-1}$ для набору даних

RetinaMNIST

Залежності $\Delta a(k)$ і $\Delta A(k)$ від $t_{i,k}$ і $\sum_{i=0}^{n-1} \frac{t_{i,k}}{n-1}$ були виміряні та зображені на

графіку для моделей, навчених на наборі даних PathMNIST (рис. 4.17). Вони демонструють дуже надійний збіг експериментально отриманої лінії інтерполяції (позначена чорною лінією) з найкращим сценарієм підвищення точності (позначеним червоною пунктирною лінією) з леми 6 і (25) в межах стандартного відхилення нахилу r .

Крім того, залежності $\Delta a(k)$ та $\Delta A(k)$ від $t_{i,k}$ і $\sum_{i=0}^{n-1} \frac{t_{i,k}}{n-1}$ для моделей, навчених

на RetinaMNIST (рис. 4.18), демонструють дуже надійний збіг експериментально отриманої лінії інтерполяції (позначена чорною лінією) з найгіршим практично можливим сценарієм підвищення точності (позначена синьою пунктирною лінією) з леми 5 і (24) в межах стандартного відхилення нахилу r .

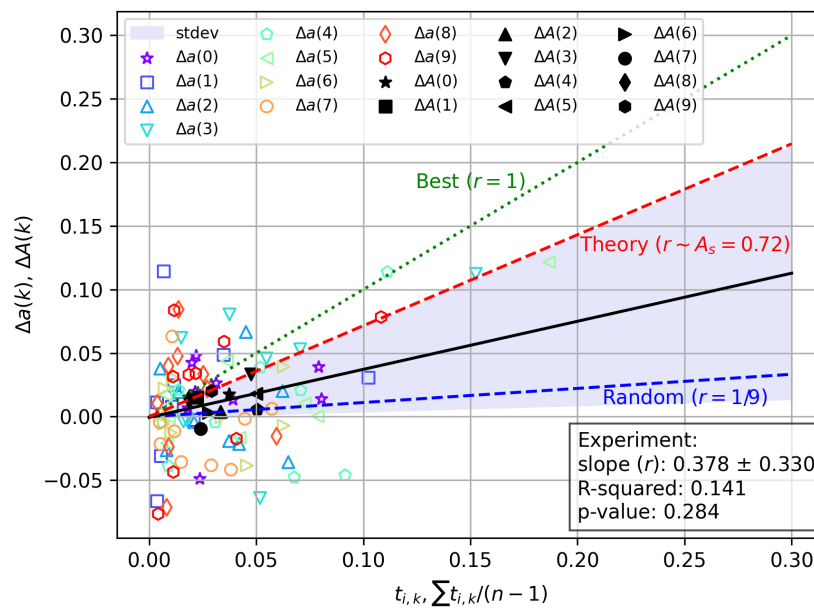


Рисунок 4.19 – Залежність $\Delta a(k)$ та $\Delta A(k)$ від $t_{i,k}$ та $\sum_{i=0}^{n-1} \frac{t_{i,k}}{n-1}$ для набору даних CIFAR10

А залежності $\Delta a(k)$ та $\Delta A(k)$ від $t_{i,k}$ і $\sum_{i=0}^{n-1} \frac{t_{i,k}}{n-1}$ для моделей, навчених на

CIFAR10 (рис. 4.19), демонструють близький збіг експериментально отриманої лінії інтерполяції (позначена чорною лінією) між найкращим і найгіршим практично можливим сценарієм підвищення точності з леми 7 і (26) в межах стандартного відхилення нахилу r .

Різні сценарії вдосконалення (для тієї самої використаної моделі глибокого навчання), проілюстровані цими експериментальними результатами, відповідають цілком різним практично можливим випадкам, коли набори даних мають різний компроміс (відповідність) між складністю даних (складними прихованими властивостями) та параметрами набору даних, такими як кількість вибірок даних, мінливість даних, гармонійне (рівне) представлення складності даних та мінливості між класами тощо. PathMNIST має відносно низьку складність даних, велику кількість вибірок та їх високу мінливість, що призводить до високих у цьому дослідженні значень точності для однієї модальності A_s рівномірно та з найкращим результатом покращення точності для мультимодальності $\Delta A(k)$. Навпаки, RetinaMNIST має відносно високу складність даних, малу кількість зразків і їх

високу варіативність, що призводить до дуже низьких у цьому дослідженні значень точності для однієї модальності A_s і до найгірших результатів покращення точності для мультимодальності $\Delta A(k)$. Відповідно, CIFAR10 має відносно високу складність даних, велику кількість вибірок та їх високу мінливість, що призводить до проміжних у цьому дослідженні значень точності для однієї модальності A_s та до проміжних результатів покращення точності для мультимодальності $\Delta A(k)$.

З фундаментальної точки зору результати дозволяють краще зрозуміти потенційне покращення точності за допомогою мультимодального доповнення метаданих та інших методів на основі аналізу, запропонованого в дусі інтерпретованого штучного інтелекту. З практичної точки зору ці результати відкривають шляхи для подальшого вдосконалення додатків штучного інтелекту на основі методів мультимодального доповнення метаданих у відповідних випадках використання. Крім того, ці результати дають ще один критерій для оцінки відповідності між складністю даних (складні приховані властивості) та параметрами набору даних (такими як кількість вибірок даних, мінливість даних, однакове представлення складності даних та мінливості між класами тощо), що може бути описовим для розуміння якості наборів даних, що використовуються на практиці.

4.5. Висновки до розділу 4

У дослідженні доповнення метаданих для багатокласової класифікації всі створені мультимодальні моделі (MP, ME, MPE) у порівнянні з одномодальною моделлю (SM) дозволили досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC для всіх класів у діапазоні від 4% до 27%, що виходять за межі стандартного відхилення 2-3% виміряного перехресною перевіркою та можуть бути оцінені як значні.

У дослідженні впливу складності мультимодальної моделі на багатокласову класифікацію мультимодальна модель (MP) у порівнянні з одномодальною моделлю

(SM) дозволила досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC для деяких класів у діапазоні від 15% до 26% (в залежності від складності архітектури згорткової компоненти нейронної мережі), що виходять за межі стандартного відхилення 3-8% виміряного перехресною перевіркою та можуть бути оцінені як значні.

Підхід, заснований на недетермінованому доповненні метаданих, а саме використання додаткових модальностей (різних текстових описів зображень) з “витоком даних” є корисною стратегією для кращої класифікації деяких важко класифікованих ступенів тяжкості діабетичної ретинопатії. У порівнянні з одномодальною моделлю (SM), мультимодальна модель (MMFE) дозволила досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC у діапазоні від 12% до 26%.

В дослідженні впливу мультимодального доповнення метаданих на точність багатокласової класифікації було показано застосовність мультимодальних моделей у контексті підвищення точності в задачах машинного навчання. Дослідження розглядало різні сценарії, починаючи від найгіршого випадку підвищення точності до найкращого сценарію, залежно від того, як помилки розподіляються між класами. Результати показали, що точність мультимодальних моделей значно змінюється залежно від складності набору даних, розміру вибірки та мінливості даних. Зокрема, мультимодальне доповнення метаданих забезпечило найбільш значні покращення точності в наборах даних із низькою складністю даних, великою кількістю вибірок і високою мінливістю даних. Навпаки, набори даних із високою складністю даних, невеликою кількістю вибірок і високою варіабельністю мали більш скромні покращення. Це дослідження дає цінну інформацію про потенціал мультимодальних підходів у інтерпретованому штучному інтелекті і підкреслює важливість характеристик набору даних у практичних застосуваннях, пропонуючи нові критерії для оцінки якості набору даних і точності моделі.

ВИСНОВКИ

В дисертаційній роботі розв'язане актуальне наукове завдання створення підходу з використанням методів комп'ютерного зору та глибоких нейронних мереж для надання сучасних передових медичних послуг на основі штучного інтелекту, спрямованих на підвищення точності багатокласової класифікації діабетичної ретинопатії. При цьому, отримано наступні наукові та практичні результати:

1. Розглянуто проблему багатокласової класифікації для одномодальної (з введенням зображення) моделі та мультимодальної (з введенням зображення та тексту) моделі та створено кілька варіантів вхідних значень і відповідних моделей на основі аналізу способів та методів використання глибокого навчання для виявлення захворювань на прикладі діабетичної ретинопатії і аналізу існуючих методологій виявлення захворювань: одномодальна модель (SM) лише з вхідним зображенням і мультимодальні моделі з вхідними зображеннями та текстом, як-от мультимодальна модель із думкою пацієнта (MP), мультимодальна модель із думкою експерта (ME), мультимодальна модель із думкою пацієнта та експерта (MPE) та мультимодальна модель з недетермінованою думкою експерта (MMFE).

2. Проведено дослідження доповнення метаданих для вирішення завдання багатокласової класифікації, яке показало, що всі створені мультимодальні моделі (MP, ME, MPE) у порівнянні з одномодальною моделлю (SM) дозволили досягти різних статистично значущих покращень точності багатокласової класифікації за значенням площі під кривою похибок (AUC) для всіх класів у діапазоні від 4% до 27%, що виходять за межі стандартного відхилення 2-3% виміряного перехресною перевіркою.

3. Проведено дослідження впливу складності мультимодальної моделі на вирішення завдання багатокласової класифікації, яке показало, що мультимодальна модель (MP) у порівнянні з одномодальною моделлю (SM) дозволила досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC для деяких класів у діапазоні від 15% до 26% (в залежності від складності архітектури згорткової компоненти

нейронної мережі), що виходять за межі стандартного відхилення 3-8% виміряного перехресною перевіркою.

4. Проведено дослідження недетермінованого доповнення метаданих для вирішення завдання багатокласової класифікації, яке показало, що мультимодальна модель (MMFE) у порівнянні з одномодальною моделлю (SM) дозволила досягти різних статистично значущих покращень точності багатокласової класифікації за значенням AUC у діапазоні від 12% до 26%.

5. Наведено теоретичні оцінки можливих (найкращих – коли точність за класами підвищується на сумарну похибку точності за окремими класами для одномодальної моделі, де похибку вдається зменшити завдяки застосуванню маркування у додатковій модальності; проміжних – коли точність за класами підвищується на сумарну похибку точності, помножену на функцію розподілу ймовірностей, де продемонстровано можливу залежність покращення від середньої точності для одномодальної моделі; і найгірших – коли точність за класами не підвищується взагалі) рівнів покращення точності багатокласової класифікації за допомогою простого математичного опису з акцентом на деяких практичних випадках.

6. На основі теоретичних оцінок проведено дослідження впливу мультимодального доповнення метаданих на точність багатокласової класифікації з використанням різних стандартних (CIFAR10) і спеціалізованих медичних (PathMNIST, RetinaMNIST) наборів даних, яке показало, що точність мультимодальних моделей значно змінюється залежно від складності набору даних, розміру вибірки та мінливості даних.

7. Запропоновано комплексний метод мультимодального аналізу медичних даних на основі глибокого навчання, який полягає у використанні “витоку даних” на крайніх і подібних класах, що дозволяє підвищити точність визначення окремих класів для вирішення завдань багатокласової класифікації.

8. Проведено аналіз результатів використання запропонованого комплексного методу мультимодального аналізу медичних даних, який показав, що запропонований метод на відміну від існуючих рішень за рахунок використання

додаткових модальностей на основі доповнення метаданих дозволив отримати підвищення точності визначення окремих класів для завдань багатокласової класифікації захворювань на 4-27% у порівнянні зі стандартним одномодальним підходом для розглянутих ідентичних наборів даних і архітектур нейронних мереж.

Методи, пов'язані зі штучним інтелектом, можуть автоматизувати та значно пришвидшити програми скринінгу шляхом автоматизованої обробки медичних даних без залучення медичного персоналу на етапі скринінгу. Одержані результати дозволяють застосовувати моделі глибоких нейронних мереж для автоматизованої обробки медичних даних в якості допоміжного інструмента для медичних працівників для виявлення та багатокласової класифікації захворювання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. King, H., Aubert, R. E., & Herman, W. H. (1998). Global burden of diabetes, 1995–2025: prevalence, numerical estimates, and projections. *Diabetes care*, 21(9), 1414-1431.
2. Rema, M., Premkumar, S., Anitha, B., Deepa, R., Pradeepa, R., & Mohan, V. (2005). Prevalence of diabetic retinopathy in urban India: the Chennai Urban Rural Epidemiology Study (CURES) eye study, I. *Investigative ophthalmology & visual science*, 46(7), 2328-2333.
3. David, A., Klein, R., & Gardner, T. (2012). Mechanisms of disease diabetic retinopathy. *N. Engl. Med. J*, 366, 1227-1239.
4. Wild, S., Roglic, G., Green, A., Sicree, R., & King, H. (2004). Global prevalence of diabetes: estimates for the year 2000 and projections for 2030. *Diabetes care*, 27(5), 1047-1053.
5. Fong, D. S., Aiello, L., Gardner, T. W., King, G. L., Blankenship, G., Cavallerano, J. D., ... & American Diabetes Association. (2004). Retinopathy in diabetes. *Diabetes care*, 27(suppl_1), s84-s87.
6. Klein, R., Klein, B. E., Moss, S. E., & Cruickshanks, K. J. (1998). The Wisconsin Epidemiologic Study of Diabetic Retinopathy: XVII: The 14-year incidence and progression of diabetic retinopathy and associated risk factors in type 1 diabetes. *Ophthalmology*, 105(10), 1801-1815.
7. Bhavsar, A. R. (2002). Diabetic retinopathy. The diabetes eye exam initiative. *Minnesota medicine*, 85(6), 46-47.
8. Matsushashi, H., Tamura, M., Kudoh, M., Suda, T., Yagihashi, S., & Nakazawa, M. (2001). Epidemiologic study of diabetic retinopathy in nine hospitals in the Aomori area. *Nippon Ganka Gakkai Zasshi*, 105(11), 760-765.
9. Varma, R., Torres, M., Peña, F., Klein, R., Azen, S. P., & Los Angeles Latino Eye Study Group. (2004). Prevalence of diabetic retinopathy in adult Latinos: the Los Angeles Latino eye study. *Ophthalmology*, 111(7), 1298-1306.
10. Massin, P., Chabouis, A., Erginay, A., Viens-Bitker, C., Lecleire-Collet, A.,

Meas, T., ... & Denormandie, P. (2008). OPHDIAT©: A telemedical network screening system for diabetic retinopathy in the Île-de-France. *Diabetes & metabolism*, 34(3), 227-234.

11. Romero-Aroca, P., Fernández-Balart, J., Baget-Bernaldiz, M., Martinez-Salcedo, I., Méndez-Marín, I., Salvat-Serra, M., & Buil-Calvo, J. A. (2009). Changes in the diabetic retinopathy epidemiology after 14 years in a population of Type 1 and 2 diabetic patients after the new diabetes mellitus diagnosis criteria and a more strict control of the patients. *Journal of diabetes and its complications*, 23(4), 229-238.

12. Chen, E., Looman, M., Laouri, M., Gallagher, M., Van Nuys, K., Lakdawalla, D., & Fortuny, J. (2010). Burden of illness of diabetic macular edema: literature review. *Current medical research and opinion*, 26(7), 1587-1597.

13. Diabetes, U. K. (2004). Diabetes in the UK. Key statistics on diabetes.

14. e Brito, C. P. (2009). Resultados preliminares do estudo de prevalência da diabetes em Portugal (PREVADIAB-2009). *Revista Portuguesa de Diabetes*, 4(2), 88-90.

15. Falcão, I. M., Pinto, C., Santos, J., Fernandes, M. D. L., Ramalho, L., Paixão, E., & Falcão, J. M. (2008). Estudo da Prevalência da diabetes e das suas complicações numa coorte de diabéticos portugueses: um estudo na Rede Médicos-Sentinela. *Revista Portuguesa de Medicina Geral e Familiar*, 24(6), 679-92.

16. Javitt, J. C., & Aiello, L. P. (1996). Cost-effectiveness of detecting and treating diabetic retinopathy. *Annals of internal medicine*, 124(1_Part_2), 164-169.

17. Snyderman, R., & Williams, R. S. (2003). Prospective medicine: the next health care transformation. *Academic Medicine*, 78(11), 1079-1084.

18. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91-110.

19. Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.

20. Ojala, T., Pietikainen, M., & Harwood, D. (1994, October). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of 12th international conference on pattern recognition (Vol.

1, pp. 582-585). IEEE.

21. Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). Ieee.

22. Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 11-15.

23. Zuluaga, M. A., Magnin, I. E., Hernández Hoyos, M., Delgado Leyton, E. J., Lozano, F., & Orkisz, M. (2011). Automatic detection of abnormal vascular cross-sections based on density level detection and support vector machines. *International journal of computer assisted radiology and surgery*, 6, 163-174.

24. Donner, R., Birngruber, E., Steiner, H., Bischof, H., & Langs, G. (2011). Localization of 3D anatomical structures using random forests and discrete optimization. In *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging: International MICCAI Workshop, MCV 2010, Beijing, China, September 20, 2010, Revised Selected Papers 1* (pp. 86-95). Springer Berlin Heidelberg.

25. Zuluaga, M. A., Delgado Leyton, E. J., Hernández Hoyos, M., & Orkisz, M. (2011). Feature selection for SVM-based vascular anomaly detection. In *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging: International MICCAI Workshop, MCV 2010, Beijing, China, September 20, 2010, Revised Selected Papers 1* (pp. 141-152). Springer Berlin Heidelberg.

26. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.

27. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

28. Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104, 154-171.

29. Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international

conference on computer vision (pp. 1440-1448).

30. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

31. Savardi, M., Benini, S., & Signoroni, A. (2018, September). -Hemolysis Detection on Cultured Blood Agar Plates by Convolutional Neural Networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II* (pp. 30-38). Cham: Springer International Publishing.

32. Setio, A. A. A., Ciompi, F., Litjens, G., Gerke, P., Jacobs, C., Van Riel, S. J., ... & Van Ginneken, B. (2016). Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. *IEEE transactions on medical imaging*, 35(5), 1160-1169.

33. Teramoto, A., Fujita, H., Yamamuro, O., & Tamaki, T. (2016). Automated detection of pulmonary nodules in PET/CT images: Ensemble false-positive reduction using a convolutional neural network technique. *Medical physics*, 43(6Part1), 2821-2827.

34. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

35. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III* 18 (pp. 234-241). Springer International Publishing.

36. Payer, C., Štern, D., Bischof, H., & Urschler, M. (2016). Regressing heatmaps for multiple landmark localization using CNNs. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II* 19 (pp. 230-238). Springer International Publishing.

37. Mader, A. O., von Berg, J., Fabritz, A., Lorenz, C., & Meyer, C. (2018).

Localization and labeling of posterior ribs in chest radiographs using a CRF-regularized FCN with local refinement. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II* 11 (pp. 562-570). Springer International Publishing.

38. Meyer, M. I., Galdran, A., Mendonça, A. M., & Campilho, A. (2018). A pixel-wise distance regression approach for joint retinal optical disc and fovea detection. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II* 11 (pp. 39-47). Springer International Publishing.

39. Kiraly, A. P., Nader, C. A., Tuysuzoglu, A., Grimm, R., Kiefer, B., El-Zehiry, N., & Kamen, A. (2017). Deep convolutional encoder-decoders for prostate cancer detection and classification. In *Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III* 20 (pp. 489-497). Springer International Publishing.

40. Dubost, F., Bortsova, G., Adams, H., Ikram, A., Niessen, W. J., Vernooij, M., & De Bruijne, M. (2017, September). Gp-unet: Lesion detection from weak labels with a 3d regression network. In *Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III* (pp. 214-221). Cham: Springer International Publishing.

41. Wang, Z., Yin, Y., Shi, J., Fang, W., Li, H., & Wang, X. (2017). Zoom-in-net: Deep mining lesions for diabetic retinopathy detection. In *Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III* 20 (pp. 267-275). Springer International Publishing.

42. Yang, X., Wang, Z., Liu, C., Le, H. M., Chen, J., Cheng, K. T., & Wang, L. (2017). Joint detection and diagnosis of prostate cancer in multi-parametric MRI based on multimodal convolutional neural networks. In *Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC,*

Canada, September 11-13, 2017, Proceedings, Part III 20 (pp. 426-434). Springer International Publishing.

43. Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9), 1627-1645.

44. Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 761-769).

45. Yang, Y., Li, T., Li, W., Wu, H., Fan, W., & Zhang, W. (2017). Lesion detection and grading of diabetic retinopathy via two-stages deep convolutional neural networks. In *Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20* (pp. 533-540). Springer International Publishing.

46. Cai, Y., Landis, M., Laidley, D. T., Kornecki, A., Lum, A., & Li, S. (2016). Multi-modal vertebrae recognition using transformed deep convolution network. *Computerized medical imaging and graphics*, 51, 11-19.

47. Li, Y., Alansary, A., Cerrolaza, J. J., Khanal, B., Sinclair, M., Matthew, J., ... & Rueckert, D. (2018). Fast multiple landmark localisation using a patch-based iterative network. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I* (pp. 563-571). Springer International Publishing.

48. Zheng, Y., Liu, D., Georgescu, B., Nguyen, H., & Comaniciu, D. (2015). 3D deep learning for efficient and robust landmark detection in volumetric data. In *Medical Image Computing and Computer-Assisted Intervention--MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part I 18* (pp. 565-572). Springer International Publishing.

49. Chen, H., Ni, D., Qin, J., Li, S., Yang, X., Wang, T., & Heng, P. A. (2015). Standard plane localization in fetal ultrasound via domain transferred deep neural networks. *IEEE journal of biomedical and health informatics*, 19(5), 1627-1636.

50. Baumgartner, C. F., Kamnitsas, K., Matthew, J., Fletcher, T. P., Smith, S.,

Koch, L. M., ... & Rueckert, D. (2017). SonoNet: real-time detection and localisation of fetal standard scan planes in freehand ultrasound. *IEEE transactions on medical imaging*, 36(11), 2204-2215.

51. Ma, C., Huang, J. B., Yang, X., & Yang, M. H. (2015). Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision* (pp. 3074-3082).

52. Kumar, A., Sridar, P., Quinton, A., Kumar, R. K., Feng, D., Nanan, R., & Kim, J. (2016, April). Plane identification in fetal ultrasound images using saliency maps and convolutional neural networks. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)* (pp. 791-794). IEEE.

53. Setio, A. A. A., Traverso, A., De Bel, T., Berens, M. S., Van Den Bogaard, C., Cerello, P., ... & Jacobs, C. (2017). Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge. *Medical image analysis*, 42, 1-13.

54. Ding, J., Li, A., Hu, Z., & Wang, L. (2017). Accurate pulmonary nodule detection in computed tomography images using deep convolutional neural networks. In *Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20* (pp. 559-567). Springer International Publishing.

55. Dou, Q., Chen, H., Jin, Y., Lin, H., Qin, J., & Heng, P. A. (2017). Automated pulmonary nodule detection via 3d convnets with online sample filtering and hybrid-loss residual learning. In *Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20* (pp. 630-638). Springer International Publishing.

56. Zhu, W., Liu, C., Fan, W., & Xie, X. (2018, March). Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification. In *2018 IEEE winter conference on applications of computer vision (WACV)* (pp. 673-681). IEEE.

57. Platania, R., Shams, S., Yang, S., Zhang, J., Lee, K., & Park, S. J. (2017, August). Automated breast cancer diagnosis using deep learning and region of interest detection (bc-droid). In *Proceedings of the 8th ACM international conference on*

bioinformatics, computational biology, and health informatics (pp. 536-543).

58. Al-Masni, M. A., Al-Antari, M. A., Park, J. M., Gi, G., Kim, T. Y., Rivera, P., ... & Kim, T. S. (2018). Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Computer methods and programs in biomedicine*, 157, 85-94.

59. Kooi, T., van Ginneken, B., Karssemeijer, N., & den Heeten, A. (2017). Discriminating solitary cysts from soft tissue lesions in mammography using a pretrained deep convolutional neural network. *Medical physics*, 44(3), 1017-1027.

60. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

61. López-Linares, K., Aranjuelo, N., Kabongo, L., Maclair, G., Lete, N., Ceresa, M., ... & Ballester, M. A. G. (2018). Fully automatic detection and segmentation of abdominal aortic thrombus in post-operative CTA images using deep convolutional neural networks. *Medical image analysis*, 46, 202-214.

62. Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., ... & Heng, P. A. (2016). Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. *IEEE transactions on medical imaging*, 35(5), 1182-1195.

63. Nair, T., Precup, D., Arnold, D. L., & Arbel, T. (2020). Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical image analysis*, 59, 101557.

64. Xu, C., Xu, L., Gao, Z., Zhao, S., Zhang, H., Zhang, Y., ... & Li, S. (2017). Direct detection of pixel-level myocardial infarction areas via a deep-learning algorithm. In *Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20* (pp. 240-249). Springer International Publishing.

65. Kuo, W., Häne, C., Yuh, E., Mukherjee, P., & Malik, J. (2018). Cost-sensitive active learning for intracranial hemorrhage detection. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part III 11* (pp. 715-723). Springer

International Publishing.

66. Asiri, N., Hussain, M., Al Adel, F., & Alzaidi, N. (2019). Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey. *Artificial intelligence in medicine*, 99, 101701.
67. Early Treatment Diabetic Retinopathy Study Research Group. (1991). Grading diabetic retinopathy from stereoscopic color fundus photographs—an extension of the modified Airlie House classification: ETDRS report number 10. *Ophthalmology*, 98(5), 786-806.
68. Acharya, U. R., Lim, C. M., Ng, E. Y. K., Chee, C., & Tamura, T. (2009). Computer-based detection of diabetes retinopathy stages using digital fundus images. *Proceedings of the institution of mechanical engineers, part H: journal of engineering in medicine*, 223(5), 545-553.
69. Kanski, J. J. (2009). *Clinical ophthalmology: a synopsis*. Elsevier Health Sciences.
70. Zachariah, S., Wykes, W., & Yorston, D. (2015). Grading diabetic retinopathy (DR) using the Scottish grading protocol. *Community eye health*, 28(92), 72.
71. Seoud, L., Hurtut, T., Chelbi, J., Cheriet, F., & Langlois, J. P. (2015). Red lesion detection using dynamic shape features for diabetic retinopathy screening. *IEEE transactions on medical imaging*, 35(4), 1116-1126.
72. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60-88.
73. Gondal, W. M., Köhler, J. M., Grzeszick, R., Fink, G. A., & Hirsch, M. (2017, September). Weakly-supervised localization of diabetic retinopathy lesions in retinal fundus images. In *2017 IEEE international conference on image processing (ICIP)* (pp. 2069-2073). IEEE.
74. Li, Q., Feng, B., Xie, L., Liang, P., Zhang, H., & Wang, T. (2015). A cross-modality learning approach for vessel segmentation in retinal images. *IEEE transactions on medical imaging*, 35(1), 109-118.
75. Arunkumar, R., & Karthigaikumar, P. (2017). Multi-retinal disease

classification by reduced deep learning features. *Neural Computing and Applications*, 28, 329-334.

76. Messidor dataset [Електронний ресурс]. – Режим доступу: <https://www.adcis.net/en/third-party/messidor/>

77. E-optha [Електронний ресурс]. – Режим доступу: <https://www.adcis.net/en/third-party/e-optha/>

78. Kaggle dataset [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/c/diabetic-retinopathy-detection/data>

79. Tymchenko, B., Marchenko, P., & Spodarets, D. (2020). Deep learning approach to diabetic retinopathy detection. *arXiv preprint arXiv:2003.02261*.

80. APTOS 2019 Виявлення сліпоти [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/c/aptos2019-blindness-detection>

81. Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1), 1-9.

82. Codella, N. C., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., ... & Halpern, A. (2018, April). Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)* (pp. 168-172). IEEE.

83. Prentašić, P., Lončarić, S., Vatauvuk, Z., Benčić, G., Subašić, M., Petković, T., ... & Tadić, R. (2013, September). Diabetic retinopathy image database (DRiDB): a new database for diabetic retinopathy screening programs research. In *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)* (pp. 711-716). IEEE.

84. Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., & Rodellar, J. (2020). A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in brief*, 30.

85. He, A., Li, T., Li, N., Wang, K., & Fu, H. (2020). CABNet: category attention block for imbalanced diabetic retinopathy grading. *IEEE Transactions on Medical Imaging*, 40(1), 143-153.

86. Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).
87. Lin, J., Cai, Q., & Lin, M. (2021). Multi-label classification of fundus images with graph convolutional network and self-supervised learning. IEEE Signal Processing Letters, 28, 454-458.
88. Eyepacs dataset [Электронный ресурс]. – Режим доступа: <http://www.eyepacs.com/solution>
89. Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
90. Li, X., Hu, X., Yu, L., Zhu, L., Fu, C. W., & Heng, P. A. (2019). CANet: cross-disease attention network for joint diabetic retinopathy and diabetic macular edema grading. IEEE transactions on medical imaging, 39(5), 1483-1493.
91. Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).
92. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
93. Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. Encyclopedia of database systems, 5, 532-538.
94. Dataset, D. C. (2020). The 2nd diabetic retinopathy–grading and image quality estimation challenge.
95. Yang, J., Shi, R., & Ni, B. (2021, April). Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI) (pp. 191-195). IEEE.
96. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., ... & Ni, B. (2023). MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification. Scientific Data, 10(1), 41.
97. EyePACS dataset [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/datasets/mariaherrerot/eyepacspreprocess>
98. Shulha, M., Gordienko, Y., & Stirenko, S. (2022, September). Impact of

multimodal model complexity on classification of diabetic retinopathy level. In *International Conference on Computing, Intelligence and Data Analytics* (pp. 168-180). Cham: Springer International Publishing.

99. Shulha, M., Gordienko, Y., & Stirenko, S. (2023, February). Deep learning with metadata augmentation for classification of diabetic retinopathy level. In *Proceedings of Third International Conference on Sustainable Expert Systems: ICSES 2022* (pp. 613-630). Singapore: Springer Nature Singapore.

100. Gordienko, Y., Shulha, M., Kochura, Y., Rokovyi, O., Alienin, O., & Stirenko, S. (2023). Fuzzy Metadata Augmentation for Multimodal Data Classification. In *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2023* (pp. 157-172). Singapore: Springer Nature Singapore.

101. Gordienko, Y., Shulha, M., Kochura, Y., Rokovyi, O., Alienin, O., Taran, V., & Stirenko, S. (2023). Ensemble Knowledge Distillation for Edge Intelligence in Medical Applications. In *Advancements in Knowledge Distillation: Towards New Horizons of Intelligent Systems* (pp. 135-168). Cham: Springer International Publishing.

102. Zago, G. T., Andreão, R. V., Dorizzi, B., & Salles, E. O. T. (2020). Diabetic retinopathy detection using red lesion localization and convolutional neural networks. *Computers in biology and medicine*, 116, 103537.

103. Liu, R., Wang, X., Wu, Q., Dai, L., Fang, X., Yan, T., ... & Zhang, P. (2022). Deepdrid: Diabetic retinopathy—grading and image quality estimation challenge. *Patterns*, 3(6).

104. Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Sahasrabuddhe, V., & Meriaudeau, F. (2018). Indian diabetic retinopathy image dataset (IDRiD): a database for diabetic retinopathy screening research. *Data*, 3(3), 25.

105. Decencière, E., Zhang, X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., ... & Klein, J. C. (2014). Feedback on a publicly distributed image database: the Messidor database. *Image Analysis & Stereology*, 33(3), 231-234.

106. Kather, J. N., Krisam, J., Charoentong, P., Luedde, T., Herpel, E., Weis, C. A., ... & Halama, N. (2019). Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1), e1002730.

107. Abràmoff, M. D., Lou, Y., Erginay, A., Clarida, W., Amelon, R., Folk, J. C., & Niemeijer, M. (2016). Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. *Investigative ophthalmology & visual science*, 57(13), 5200-5206.
108. Zhang, W., Zhong, J., Yang, S., Gao, Z., Hu, J., Chen, Y., & Yi, Z. (2019). Automated identification and grading system of diabetic retinopathy using deep neural networks. *Knowledge-Based Systems*, 175, 12-25.
109. Wang, J., Luo, J., Liu, B., Feng, R., Lu, L., & Zou, H. (2020). Automated diabetic retinopathy grading and lesion detection based on the modified R-FCN object-detection algorithm. *IET Computer Vision*, 14(1), 1-8.
110. Pao, S. I., Lin, H. Z., Chien, K. H., Tai, M. C., Chen, J. T., & Lin, G. M. (2020). Detection of diabetic retinopathy using bichannel convolutional neural network. *Journal of Ophthalmology*, 2020.

ДОДАТОК А

Код реалізації програмного засобу для проведення дослідження доповнення метаданих для вирішення завдання багатокласової класифікації

Лістинг А.1 – Компонента обробки вхідних даних

```
# Install medmnist as a standard Python package
! pip install medmnist

# Import libraries
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.text import text_to_word_sequence
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Concatenate
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Model
import numpy as np
import matplotlib.pyplot as plt
import logging
tf.get_logger().setLevel(logging.ERROR)
from tqdm import tqdm
import medmnist
from medmnist import INFO, Evaluator
import torch.utils.data as data
import torchvision.transforms as transforms
from tensorflow.keras.callbacks import ModelCheckpoint
import os, sys, pathlib
from sklearn.metrics import confusion_matrix
from sklearn.metrics import RocCurveDisplay
from sklearn.model_selection import KFold
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2
from tensorflow.keras.applications.efficientnet import EfficientNetB0
from tensorflow.keras.applications import ResNet50

# Global parameters
MODEL_NAME = 'ResNet50'
SIZE = 32
WEIGHTS = None
MODALITY = 'SINGLE'
BATCH_SIZE = 128

# Prepare dataset - MedMNIST
data_flag = 'retinamnist'
download = True
info = INFO[data_flag]
task = info['task']
n_channels = info['n_channels']
n_classes = len(info['label'])
DataClass = getattr(medmnist, info['python_class'])

# Read the MedMNIST data, preprocess and encapsulate into dataloader form
# preprocessing
```

```

data_transform = transforms.Compose([transforms.ToTensor()])

# load the data
train_dataset = DataClass(split='train', transform=data_transform, download=download)
valid_dataset = DataClass(split='val', transform=data_transform, download=download)
test_dataset = DataClass(split='test', transform=data_transform, download=download)

# encapsulate data into dataloader form
train_loader = data.DataLoader(dataset=train_dataset, batch_size=BATCH_SIZE,
                                shuffle=True)
train_loader_at_eval = data.DataLoader(dataset=valid_dataset, batch_size=BATCH_SIZE,
                                         shuffle=False)
test_loader = data.DataLoader(dataset=test_dataset, batch_size=2*BATCH_SIZE,
                               shuffle=False)

# MedMNIST - Pre-processing
train_images = train_dataset.imgs.astype(np.float32)
train_labels = train_dataset.labels.astype(np.float32)
valid_images = valid_dataset.imgs.astype(np.float32)
valid_labels = valid_dataset.labels.astype(np.float32)
test_images = test_dataset.imgs.astype(np.float32)
test_labels = test_dataset.labels.astype(np.float32)

# Standardize the data.
mean = np.mean(train_images)
stddev = np.std(train_images)

train_images = (train_images - mean) / stddev
valid_images = (valid_images - mean) / stddev
test_images = (test_images - mean) / stddev

import skimage
from skimage.transform import resize

target_size = (SIZE, SIZE, 3)
train_images_resized = np.asarray([skimage.transform.resize(image, target_size) for
image in train_images])
valid_images_resized = np.asarray([skimage.transform.resize(image, target_size) for
image in valid_images])
test_images_resized = np.asarray([skimage.transform.resize(image, target_size) for
image in test_images])

# Directories, files, ...
N_SPLITS = 6
kFold = KFold(n_splits=N_SPLITS)
DA_TYPE = 'NODA'
EPOCHS = 100
num_classes = 5
SAVED_MODEL = 'MAX_AUC'
DATASET = data_flag

model_filename_prefix = MODEL_NAME + '_' + str(SIZE) + '_' + str(WEIGHTS) + '_' +
MODALITY + '_' + DA_TYPE + '_' + DATASET + '_'

# Path to the BASE folder with results.
base_dir = model_filename_prefix + 'RESULTS/'

# Path to the subfolder with FIGURES of results.
figures_dir = base_dir + 'FIGURES/'
pathlib.Path(figures_dir).mkdir(parents=True, exist_ok=True)

# Path to the subfolder with MODELS of results.
models_dir = base_dir + 'MODELS/'
pathlib.Path(models_dir).mkdir(parents=True, exist_ok=True)

```

```

# Path to the subfolder with LOGS of results - RESERVED for future ...
tb_main_log_dir = base_dir + 'LOGS'
pathlib.Path(tb_main_log_dir).mkdir(parents=True, exist_ok=True)

max_AUC_model_filename = model_filename_prefix + 'MAX-AUC.hdf5'
max_AUC_model_filepath = os.path.join(models_dir, max_AUC_model_filename)
max_acc_model_filename = model_filename_prefix + 'MAX-ACC.hdf5'
max_acc_model_filepath = os.path.join(models_dir, max_acc_model_filename)
min_loss_model_filename = model_filename_prefix + 'MAX-LOSS.hdf5'
min_loss_model_filepath = os.path.join(models_dir, min_loss_model_filename)

#Modality 2 - Expert text
MAX_WORDS = 2
EMBEDDING_WIDTH = 1

# Function to create second modality.
def create_text_2(tokenizer, labels):
    text = []

    for i, label in enumerate(labels):
        if label > 0:
            # diseased
            text.append('yes')
        else:
            # healthy
            text.append('no')

    text = tokenizer.texts_to_sequences(text)
    text = pad_sequences(text)
    return text

# Create second modality for training and test set.
vocabulary = ['no', 'yes']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

train_text = create_text_2(tokenizer, train_labels)
valid_text = create_text_2(tokenizer, valid_labels)
test_text = create_text_2(tokenizer, test_labels)

if MODALITY == 'SINGLE':
    X, Y = train_images_resized, train_labels
else:
    if MODALITY == 'MULTI':
        X, T, Y = train_images_resized, train_text, train_labels
    else:
        sys.exit('Error message: Wrong mode for SINGLE-MULTI modality.')

```

Лістинг A.2 - Компонента для роботи нейронної мережі

```

#Create models
def create_model_single():
    target_size = (SIZE, SIZE, 3)
    TRAINABLE = True

    base_model = ResNet50(
        include_top=False,
        pooling='max',
        weights=WEIGHTS,
        input_shape = target_size
    )
    # make the weights and biases of the base model non-trainable

```

```

# by "freezing" each layer of the BASE network
for layer in base_model.layers:
    layer.trainable = TRAINABLE

# Declare layers - IMAGE modality
flat_layer = tf.keras.layers.Flatten()
dense1_layer = tf.keras.layers.Dense(512, activation='relu')
dense2_layer = tf.keras.layers.Dense(5, activation='softmax')

# Connect layers - IMAGE modality
image_input = Input(shape=target_size)
x = base_model(image_input)
x = flat_layer(x)
x = dense1_layer(x)
outputs = dense2_layer(x)

model = Model(image_input, outputs)

METRICS = ['acc', keras.metrics.AUC(name='auc')]
model.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
optimizer=tf.keras.optimizers.Adam(), metrics=METRICS)

return model

def create_model_multi():

    target_size = (SIZE, SIZE, 3)
    TRAINABLE = True

    image_input = Input(shape=target_size)
    text_input = Input(shape=(EMBEDDING_WIDTH, ))

    base_model = ResNet50(
        include_top=False,
        pooling='max',
        weights=WEIGHTS,
        input_shape = target_size
    )
    # make the weights and biases of the base model non-trainable
    # by "freezing" each layer of the BASE network
    for layer in base_model.layers:
        layer.trainable = TRAINABLE

    # Declare layers - TEXT modality
    embedding_layer = Embedding(output_dim=EMBEDDING_WIDTH, input_dim = MAX_WORDS)
    lstm_layer = LSTM(8)
    concat_layer = Concatenate()

    # Declare layers - IMAGE modality
    flat_layer = tf.keras.layers.Flatten()
    dense1_layer = tf.keras.layers.Dense(512, activation='relu')
    dense2_layer = tf.keras.layers.Dense(5, activation='softmax')

    # Connect layers - TEXT modality
    embedding_output = embedding_layer(text_input)
    lstm_output = lstm_layer(embedding_output)

    # Connect layers - IMAGE modality
    x = base_model(image_input)
    x = flat_layer(x)

    # Concatenate modalities
    x = concat_layer([x, lstm_output])

```

```

# Prepare output
x = dense1_layer(x)
outputs = dense2_layer(x)

model = Model([image_input, text_input], outputs)

METRICS = ['acc',keras.metrics.AUC(name='auc')]
model.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
optimizer=tf.keras.optimizers.Adam(), metrics=METRICS)

return model

# Training - CV
def train_evaluate(model, x_train, y_train, x_val, y_val, x_test, y_test):

    y_train = tf.keras.utils.to_categorical(y_train, num_classes=num_classes)
    y_val = tf.keras.utils.to_categorical(y_val, num_classes=num_classes)
    y_test = tf.keras.utils.to_categorical(y_test, num_classes=num_classes)

    history = model.fit(x_train, y_train, epochs=EPOCHS, validation_data = (x_val,
y_val), callbacks = callbacks_list, batch_size=BATCH_SIZE, verbose=2)

    # load weights
    if SAVED_MODEL == 'MAX_ACC':
        model.load_weights(max_acc_model_filepath)
    else:
        if SAVED_MODEL == 'MIN_LOSS':
            model.load_weights(min_loss_model_filepath)
        else:
            model.load_weights(max_AUC_model_filepath)

    score = model.evaluate(x_test, y_test)
    predicted_probs = model.predict(x_test)

    predicted = predicted_probs.argmax(axis=-1)

    expected = y_test.argmax(axis=-1)

    conf_matrix = confusion_matrix(expected, predicted)
    print(conf_matrix)

    return history.history['acc'], history.history['loss'],
history.history['val_acc'], history.history['val_loss'], score[0], score[1],
conf_matrix, expected, predicted_probs

from sklearn.metrics import roc_curve, auc

scores_loss_list = []
scores_accuracy_list = []
history_acc_list = []
history_loss_list = []
history_val_acc_list = []
history_val_loss_list = []
confusion_matrix_list = []

tprs = [None] * num_classes
aucs = [None] * num_classes

for k in range(num_classes):
    tprs[k] = []
    aucs[k] = []

mean_fpr = np.linspace(0, 1, 100)

```

```

fpr_list = []
tpr_list = []
roc_auc_list = []

for i, (train, val) in tqdm(enumerate(kFold.split(X, Y))):
    fpr = dict()
    tpr = dict()
    roc_auc = dict()

    checkpoint_AUC = ModelCheckpoint(max_AUC_model_filepath, monitor='val_auc',
    verbose=1, save_best_only=True, mode='max')
    checkpoint_loss = ModelCheckpoint(min_loss_model_filepath, monitor='val_loss',
    verbose=1, save_best_only=True, mode='min')
    callbacks_list = [checkpoint_AUC, checkpoint_loss]

    if MODALITY == 'SINGLE':
        X_train_mode = X[train]
        X_valid_mode = X[val]
        X_test_mode = test_images_resized
        model = create_model_single()
    else:
        if MODALITY == 'MULTI':
            X_train_mode = [X[train], T[train]]
            X_valid_mode = [X[val], T[val]]
            X_test_mode = [test_images_resized, test_text]
            model = create_model_multi()
        else:
            sys.exit('Error message: Wrong X_train_mode for SINGLE-MULTI modality.')

    if i==0: model.summary()

    history_acc, history_loss, history_val_acc, history_val_loss, scores_loss,
    scores_accuracy, conf_matrix, expected, predicted_probs = train_evaluate(model,
    X_train_mode, Y[train], X_valid_mode, Y[val], X_test_mode, test_labels)

    scores_loss_list.append(scores_loss)
    scores_accuracy_list.append(scores_accuracy)
    history_acc_list.append(history_acc)
    history_loss_list.append(history_loss)
    history_val_acc_list.append(history_val_acc)
    history_val_loss_list.append(history_val_loss)
    confusion_matrix_list.append(conf_matrix)

    expected = tf.keras.utils.to_categorical(test_labels, num_classes=num_classes)
    for k in range(num_classes):
        viz_fpr, viz_tpr, _ = roc_curve(expected[:, k], predicted_probs[:, k])
        viz_roc_auc = auc(viz_fpr, viz_tpr)

        interp_tpr = np.interp(mean_fpr, viz_fpr, viz_tpr)
        interp_tpr[0] = 0.0
        tprs[k].append(interp_tpr)
        aucs[k].append(viz_roc_auc)

    viz_micro_fpr, viz_micro_tpr, _ = roc_curve(expected.ravel(),
    predicted_probs.ravel())
    viz_micro_roc_auc = auc(viz_micro_fpr, viz_micro_tpr)

    interp_tpr_micro = np.interp(mean_fpr, viz_micro_fpr, viz_micro_tpr)
    interp_tpr_micro[0] = 0.0
    fpr_list.append(viz_micro_fpr)
    tpr_list.append(interp_tpr_micro)
    roc_auc_list.append(viz_micro_roc_auc)

```

Training - OoCV

```

scores_loss_list = []
scores_accuracy_list = []
history_acc_list = []
history_loss_list = []
history_val_acc_list = []
history_val_loss_list = []
confusion_matrix_list = []

tprs = [None] * num_classes
aucs = [None] * num_classes

for k in range(num_classes):
    tprs[k] = []
    aucs[k] = []

mean_fpr = np.linspace(0, 1, 100)

fpr_list = []
tpr_list = []
roc_auc_list = []
fpr = dict()
tpr = dict()
roc_auc = dict()

checkpoint_AUC = ModelCheckpoint(max_AUC_model_filepath, monitor='val_auc', verbose=1,
save_best_only=True, mode='max')
checkpoint_loss = ModelCheckpoint(min_loss_model_filepath, monitor='val_loss',
verbose=1, save_best_only=True, mode='min')
callbacks_list = [checkpoint_AUC, checkpoint_loss]

if MODALITY == 'SINGLE':
    X_train_mode = train_images_resized
    X_valid_mode = valid_images_resized
    X_test_mode = test_images_resized
    model = create_model_single()
else:
    if MODALITY == 'MULTI':
        X_train_mode = [train_images_resized, train_text]
        X_valid_mode = [valid_images_resized, valid_text]
        X_test_mode = [test_images_resized, test_text]
        model = create_model_multi()
    else:
        sys.exit('Error message: Wrong X_train_mode for SINGLE-MULTI modality.')

if i==0: model.summary()

history_acc, history_loss, history_val_acc, history_val_loss, scores_loss,
scores_accuracy, conf_matrix, expected, predicted_probs = train_evaluate(model,
X_train_mode, train_labels, X_valid_mode, valid_labels, X_test_mode, test_labels)

scores_loss_list.append(scores_loss)
scores_accuracy_list.append(scores_accuracy)
history_acc_list.append(history_acc)
history_loss_list.append(history_loss)
history_val_acc_list.append(history_val_acc)
history_val_loss_list.append(history_val_loss)
confusion_matrix_list.append(conf_matrix)

expected = tf.keras.utils.to_categorical(test_labels, num_classes=num_classes)
for k in range(num_classes):
    viz_fpr, viz_tpr, _ = roc_curve(expected[:, k], predicted_probs[:, k])
    viz_roc_auc = auc(viz_fpr, viz_tpr)

    interp_tpr = np.interp(mean_fpr, viz_fpr, viz_tpr)

```

```

interp_tpr[0] = 0.0
tprs[k].append(interp_tpr)
aucs[k].append(viz_roc_auc)

viz_micro_fpr, viz_micro_tpr, _ = roc_curve(expected.ravel(), predicted_probs.ravel())
viz_micro_roc_auc = auc(viz_micro_fpr, viz_micro_tpr)

interp_tpr_micro = np.interp(mean_fpr, viz_micro_fpr, viz_micro_tpr)
interp_tpr_micro[0] = 0.0
fpr_list.append(viz_micro_fpr)
tpr_list.append(interp_tpr_micro)
roc_auc_list.append(viz_micro_roc_auc)

```

Лістинг А.3 – Компонента для обробки результатів

```

#History - Plots - CV
def history_stat_metrics(history_label, history_list):
    history_list_means = np.mean(np.asarray(history_list),axis=0)
    history_list_stds = np.std(np.asarray(history_list),axis=0)
    print('History' + history_label + ': ', history_list_means, '+-',
history_list_stds)
    return history_list_means, history_list_stds

history_train_acc_list_means, history_train_acc_list_stds = history_stat_metrics('
(accuracy - train)', history_acc_list)
history_val_acc_list_means, history_val_acc_list_stds = history_stat_metrics('
(accuracy - validation)', history_val_acc_list)

history_train_loss_list_means, history_train_loss_list_stds = history_stat_metrics('
(loss - train)', history_loss_list)
history_val_loss_list_means, history_val_loss_list_stds = history_stat_metrics(' (loss
- validation)', history_val_loss_list)

def plot_mean_std(label, line_color, fill_color, epoch_list, mean_list, std_list):
    ax.plot(
        epoch_list,
        mean_list,
        color=line_color,
        label=label,
        lw=2,
        alpha=0.8,
    )

    upper = (mean_list + std_list)
    lower = (mean_list - std_list)

    ax.fill_between(
        epoch_list,
        lower,
        upper,
        color=fill_color,
        alpha=0.5,
        label=r"$\pm$ 1 std. dev.",
    )

def plot_train_val_mean_std(metric_label, legend_location, metric_ylim,
history_train_list_means, history_train_list_stds, history_val_list_means,
history_val_list_stds):

    epochs_list = range(EPOCHS)

    plot_mean_std('train', 'r', "lightcoral", epochs_list, history_train_list_means,
history_train_list_stds)
    plot_mean_std('val', 'b', "lightsteelblue", epochs_list, history_val_list_means,

```

```

history_val_list_stds)

    ax.set(
        xlim = [0, EPOCHS-1],
        ylim = metric_ylim,
        title = metric_label + " - History",
    )

    ax.legend(loc=legend_location, ncol=2)
    fig_filename = metric_label + '.png'
    fig_file_path = os.path.join(figures_dir, fig_filename)
    plt.savefig(fig_file_path, dpi=300)
    fig_filename = metric_label + '.eps'
    fig_file_path = os.path.join(figures_dir, fig_filename)
    plt.savefig(fig_file_path, dpi=300)

plt.show()

fig, ax = plt.subplots()

plot_train_val_mean_std(model_filename_prefix + SAVED_MODEL + '_Accuracy_history',
    "lower center", [0.01, 1.05], history_train_acc_list_means,
    history_train_acc_list_stds, history_val_acc_list_means, history_val_acc_list_stds)

fig, ax = plt.subplots()

plot_train_val_mean_std(model_filename_prefix + SAVED_MODEL + '_Loss_history', "upper
center", [0.01, 2.0], history_train_loss_list_means, history_train_loss_list_stds,
    history_val_loss_list_means, history_val_loss_list_stds)

#Confusion matrix - Plots - CV
confusion_matrix_means, confusion_matrix_stds = history_stat_metrics(' (confusion
matrix)', confusion_matrix_list)

import itertools
def plot_confusion_matrix(cm, cm_std, classes,
                        normalize=False,
                        title='Means',
                        cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm_sum = cm.sum(axis=1)[:, np.newaxis]
        cm = cm.astype('float') / cm_sum
        cm_std = cm_std.astype('float') / cm_sum
        print("Confusion matrix, NORMALIZED")
    else:
        print('Confusion matrix, WITHOUT normalization')

    print('Mean values:')
    print(cm)

    print('Standard deviation values:')
    print(cm_std)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45, fontsize = 12)
    plt.yticks(tick_marks, classes, fontsize = 12)

```

```

    fmt = '.2f' if normalize else '.0f' # 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt) + "\n" + r'$\pm$' + format(cm_std[i, j],
fmt), horizontalalignment="center", color="white" if cm[i, j] > thresh else "black",
fontsize = 10)

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

    fig_filename = title + '.png'
    fig_file_path = os.path.join(figures_dir, fig_filename)
    plt.savefig(fig_file_path, dpi=300)
    fig_filename = title + '.eps'
    fig_file_path = os.path.join(figures_dir, fig_filename)
    plt.savefig(fig_file_path, dpi=300)

# Compute confusion matrix
np.set_printoptions(precision=4)

plt.figure(dpi=300)
plot_confusion_matrix(confusion_matrix_means, confusion_matrix_stds,
classes=class_names, title = model_filename_prefix + SAVED_MODEL +
'_ConfusionMatrix_NOT_normalized')

plt.figure(dpi=300)
plot_confusion_matrix(confusion_matrix_means, confusion_matrix_stds,
classes=class_names, normalize=True, title = model_filename_prefix + SAVED_MODEL +
'_ConfusionMatrix_Normalized')
plt.show()

# ROC/AUC - Plots - CV
import matplotlib._color_data as mcd

fig, ax = plt.subplots(figsize=(5, 5))

ax.plot([0, 1], [0, 1], linestyle="--", lw=2, color="black", label="Random, 0.5",
alpha=0.8)

class_names = ['0', '1', '2', '3', '4']

color_names = []
for name in mcd.CSS4_COLORS:
    color_names.append(name)

color_names = ['black', 'red', 'blue', 'green', 'cyan']
print(color_names)

for k in range(num_classes):
    mean_tpr = np.mean(tprs[k], axis=0)
    mean_tpr[-1] = 1.0
    mean_auc = auc(mean_fpr, mean_tpr)
    std_auc = np.std(aucs[k])
    ax.plot(
        mean_fpr,
        mean_tpr,
        color = color_names[k],
        label = class_names[k] + r", %0.3f $\pm$ %0.2f" % (mean_auc, std_auc),
        lw=2,
        alpha=0.8,
    )

std_tpr = np.std(tprs[k], axis=0)

```

```

    tprs_upper = np.minimum(mean_tpr + std_tpr, 1)
    tprs_lower = np.maximum(mean_tpr - std_tpr, 0)

ax.set(
    xlim=[-0.05, 1.05],
    ylim=[-0.05, 1.05],
    title = MODEL_NAME + '_' + MODALITY + '_' + DA_TYPE + '_' + SAVED_MODEL
)

# MACRO - AUC - averaged over K-folds
# Compute macro-average ROC curve and ROC area
fpr = dict()
tpr = dict()
roc_auc = dict()

# First aggregate all false positive rates
all_fpr = mean_fpr

# Then interpolate all ROC curves at this points
mean_auc_k_list = []
mean_tpr_classes = np.zeros_like(all_fpr)
for k in range(num_classes):
    mean_tpr_k = np.mean(tprs[k], axis=0)
    mean_fpr_k = mean_fpr
    mean_tpr_classes += np.interp(all_fpr, mean_fpr_k, mean_tpr_k)

    mean_auc_k = auc(mean_fpr_k, mean_tpr_k)
    mean_auc_k_list.append(mean_auc_k)

# Finally average it and compute AUC
mean_tpr_classes /= num_classes
mean_auc_classes = np.mean(mean_auc_k_list)
print(mean_auc_classes)
std_auc_classes = np.std(mean_auc_k_list)

fpr["macro"] = all_fpr
tpr["macro"] = mean_tpr_classes
roc_auc["macro"] = auc(fpr["macro"], tpr["macro"])
print(roc_auc["macro"])

ax.plot(fpr["macro"], tpr["macro"], label = "macro, %0.3f $\pm$ %0.3f" %
        (roc_auc["macro"], std_auc_classes), color='navy', linestyle=':', linewidth=4)

# MICRO - AUC - averaged over K-folds
mean_tpr_micro = np.mean(tpr_list, axis=0)
mean_roc_auc_micro = np.mean(roc_auc_list, axis=0)
std_roc_auc_micro = np.std(roc_auc_list, axis=0)
ax.plot(mean_fpr, mean_tpr_micro, label = "micro, %0.3f $\pm$ %0.3f" %
        (mean_roc_auc_micro, std_roc_auc_micro), color='deeppink', linestyle=':', linewidth=4)
ax.legend(loc="lower right", title = 'AUC (mean $\pm$ std)')

fig_filename = model_filename_prefix + SAVED_MODEL + '_ROC.png'
fig_file_path = os.path.join(figures_dir, fig_filename)
plt.savefig(fig_file_path, dpi=300)
fig_filename = model_filename_prefix + SAVED_MODEL + '_ROC.eps'
fig_file_path = os.path.join(figures_dir, fig_filename)
plt.savefig(fig_file_path, dpi=300)
plt.show()

#History - Plots - OoCV
def plot_history(label, line_color, epoch_list, metric_list):
    ax.plot(
        epoch_list,
        metric_list,

```

```

        color=line_color,
        label=label,
        lw=2,
        alpha=0.8,
    )

def plot_train_val_history(metric_title, plot_title, legend_location, metric_ylim,
                           history_train_list, history_val_list):

    epochs_list = range(EPOCHS)

    plot_history('train', 'r', epochs_list, history_train_list)
    plot_history('val', 'b', epochs_list, history_val_list)

    ax.set(
        xlim = [0, EPOCHS-1],
        ylim = metric_ylim,
        xlabel = 'Epochs',
        ylabel = metric_title,
        title = plot_title,
    )

    ax.legend(loc=legend_location, ncol=2)
    fig_filename = plot_title + '_OoF.png'
    fig_file_path = os.path.join(figures_dir, fig_filename)
    plt.savefig(fig_file_path, dpi=300)
    fig_filename = plot_title + '_OoF.eps'
    fig_file_path = os.path.join(figures_dir, fig_filename)
    plt.savefig(fig_file_path, dpi=300)

    plt.show()

fig, ax = plt.subplots()
metric_title = 'Accuracy'
plot_train_val_history(metric_title, model_filename_prefix + SAVED_MODEL + '_' +
                        metric_title + '_history', "lower center", [0.01, 1.05], history_acc, history_val_acc)

fig, ax = plt.subplots()
metric_title = 'Loss'
plot_train_val_history(metric_title, model_filename_prefix + SAVED_MODEL + '_' +
                        metric_title + '_history', "upper center", [0.01, 4.5], history_loss,
                        history_val_loss)

#Confusion matrix - Plots - OoCV
import itertools
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Means',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm_sum = cm.sum(axis=1)[:, np.newaxis]
        cm = cm.astype('float') / cm_sum
        print("Confusion matrix, NORMALIZED")
    else:
        print('Confusion matrix, WITHOUT normalization')

    print('OoF values:')
    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)

```

```

plt.title(title)
plt.colorbar()
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45, fontsize = 12)
plt.yticks(tick_marks, classes, fontsize = 12)

fmt = '.2f' if normalize else '.0f'
thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    plt.text(j, i, format(cm[i, j], fmt), horizontalalignment="center",
color="white" if cm[i, j] > thresh else "black", fontsize = 10) # fontsize = 7

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')

fig_filename = title + '_OoF.png'
fig_file_path = os.path.join(figures_dir, fig_filename)
plt.savefig(fig_file_path, dpi=300)
fig_filename = title + '_OoF.eps'
fig_file_path = os.path.join(figures_dir, fig_filename)
plt.savefig(fig_file_path, dpi=300)

class_names = ['0', '1', '2', '3', '4']
np.set_printoptions(precision=4)

plt.figure(dpi=300)
plot_confusion_matrix(confusion_matrix_means, classes=class_names, title =
model_filename_prefix + SAVED_MODEL + '_ConfusionMatrix_NOT_normalized')

plt.figure(dpi=300)
plot_confusion_matrix(confusion_matrix_means, classes=class_names, normalize=True,
title = model_filename_prefix + SAVED_MODEL + '_ConfusionMatrix_Normalized')
plt.show()

# ROC/AUC - Plots - OoCV
import matplotlib._color_data as mcd

fig, ax = plt.subplots(figsize=(5, 5))

ax.plot([0, 1], [0, 1], linestyle="--", lw=2, color="black", label="Random, 0.5",
alpha=0.8)

class_names = ['0', '1', '2', '3', '4']
color_names = []
for name in mcd.CSS4_COLORS:
    color_names.append(name)

color_names = ['black', 'red', 'blue', 'green', 'cyan']
print(color_names)

for k in range(num_classes):
    mean_tpr = np.mean(tprs[k], axis=0)
    mean_tpr[-1] = 1.0
    mean_auc = auc(mean_fpr, mean_tpr)
    ax.plot(
        mean_fpr,
        mean_tpr,
        color = color_names[k],
        label = class_names[k] + r", %0.3f" % (mean_auc),
        lw=2,
        alpha=0.8,
    )

```

```

ax.set(
    xlim=[-0.05, 1.05],
    ylim=[-0.05, 1.05],
    title = MODEL_NAME + '_' + MODALITY + '_' + DA_TYPE + '_' + SAVED_MODEL + '_OoF'
)

# MACRO - AUC - averaged over K-folds
# Compute macro-average ROC curve and ROC area
fpr = dict()
tpr = dict()
roc_auc = dict()

# First aggregate all false positive rates
all_fpr = mean_fpr

# Then interpolate all ROC curves at this points
mean_auc_k_list = []
mean_tpr_classes = np.zeros_like(all_fpr)
for k in range(num_classes):
    mean_tpr_k = np.mean(tprs[k], axis=0)
    mean_fpr_k = mean_fpr
    mean_tpr_classes += np.interp(all_fpr, mean_fpr_k, mean_tpr_k)

    mean_auc_k = auc(mean_fpr_k, mean_tpr_k)
    mean_auc_k_list.append(mean_auc_k)

# Finally average it and compute AUC
mean_tpr_classes /= num_classes
mean_auc_classes = np.mean(mean_auc_k_list)
print(mean_auc_classes)

fpr["macro"] = all_fpr
tpr["macro"] = mean_tpr_classes
roc_auc["macro"] = auc(fpr["macro"], tpr["macro"])
print(roc_auc["macro"])

ax.plot(fpr["macro"], tpr["macro"],
        label = "macro, %0.3f" % (roc_auc["macro"]),
        color='navy', linestyle=':', linewidth=4)

# MICRO - AUC - averaged over K-folds
mean_tpr_micro = np.mean(tpr_list, axis=0)
mean_roc_auc_micro = np.mean(roc_auc_list, axis=0)
ax.plot(mean_fpr, mean_tpr_micro,
        label = "micro, %0.3f" % (mean_roc_auc_micro),
        color='deeppink', linestyle=':', linewidth=4)
ax.legend(loc="lower right", title = 'AUC')

fig_filename = model_filename_prefix + SAVED_MODEL + '_OoF_ROC.png'
fig_file_path = os.path.join(figures_dir, fig_filename)
plt.savefig(fig_file_path, dpi=300)
fig_filename = model_filename_prefix + SAVED_MODEL + '_OoF_ROC.eps'
fig_file_path = os.path.join(figures_dir, fig_filename)
plt.savefig(fig_file_path, dpi=300)
plt.show()

```

ДОДАТОК Б

Фрагменти коду реалізації компоненти обробки вхідних даних програмного засобу для проведення дослідження недетермінованого доповнення метаданих для вирішення завдання багатокласової класифікації

Лістинг Б.1 – Зразок коду для створення текстової модальності згідно з шаблонами описаними в розділі 3.3.2

```
#Modality 2 - FUZZY_1_2_2 template

MAX_WORDS = 3
EMBEDDING_WIDTH = 1

# Function to create second modality.
def create_text_FUZZY_1_2_2(tokenizer, labels):
    text = []

    for label in labels:
        if label == 0:
            text.append('light')
        if label == 1:
            text.append('medium')
        if label == 2:
            text.append('medium')
        if label == 3:
            text.append('heavy')
        if label == 4:
            text.append('heavy')

    text = tokenizer.texts_to_sequences(text)
    text = pad_sequences(text)
    return text

# Create second modality for training and test set.
vocabulary = ['light', 'medium', 'heavy']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

train_text = create_text_FUZZY_1_2_2(tokenizer, train_labels)
valid_text = create_text_FUZZY_1_2_2(tokenizer, valid_labels)
test_text = create_text_FUZZY_1_2_2(tokenizer, test_labels)

#Modality 2 - FUZZY_1_3_1 template

MAX_WORDS = 3
EMBEDDING_WIDTH = 1

# Function to create second modality.
def create_text_FUZZY_1_3_1(tokenizer, labels):
    text = []

    for label in labels:
        if label == 0:
            text.append('light')
        if label == 1:
            text.append('medium')
        if label == 2:
            text.append('medium')
```

```

    if label == 3:
        text.append('medium')
    if label == 4:
        text.append('heavy')

text = tokenizer.texts_to_sequences(text)
text = pad_sequences(text)
return text

```

```

# Create second modality for training and test set.
vocabulary = ['light', 'medium', 'heavy']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

train_text = create_text_FUZZY_1_3_1(tokenizer, train_labels)
valid_text = create_text_FUZZY_1_3_1(tokenizer, valid_labels)
test_text = create_text_FUZZY_1_3_1(tokenizer, test_labels)

```

#Modality 2 - FUZZY_1_4 template

```

MAX_WORDS = 2
EMBEDDING_WIDTH = 1

# Function to create second modality.
def create_text_FUZZY_1_4(tokenizer, labels):
    text = []

    for label in labels:
        if label == 0:
            text.append('healthy')
        if label == 1:
            text.append('ill')
        if label == 2:
            text.append('ill')
        if label == 3:
            text.append('ill')
        if label == 4:
            text.append('ill')

    text = tokenizer.texts_to_sequences(text)
    text = pad_sequences(text)
    return text

```

```

# Create second modality for training and test set.
vocabulary = ['healthy', 'ill']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

train_text = create_text_FUZZY_1_4(tokenizer, train_labels)
valid_text = create_text_FUZZY_1_4(tokenizer, valid_labels)
test_text = create_text_FUZZY_1_4(tokenizer, test_labels)

```

#Modality 2 - FUZZY_2_2_1 template

```

MAX_WORDS = 3
EMBEDDING_WIDTH = 1

# Function to create second modality.
def create_text_FUZZY_2_2_1(tokenizer, labels):
    text = []

    for label in labels:
        if label == 0:
            text.append('light')

```

```

    if label == 1:
        text.append('light')
    if label == 2:
        text.append('medium')
    if label == 3:
        text.append('medium')
    if label == 4:
        text.append('heavy')

text = tokenizer.texts_to_sequences(text)
text = pad_sequences(text)
return text

```

```

# Create second modality for training and test set.
vocabulary = ['light', 'medium', 'heavy']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

train_text = create_text_FUZZY_2_2_1(tokenizer, train_labels)
valid_text = create_text_FUZZY_2_2_1(tokenizer, valid_labels)
test_text = create_text_FUZZY_2_2_1(tokenizer, test_labels)

```

#Modality 2 - FUZZY_2_3 template

```

MAX_WORDS = 2
EMBEDDING_WIDTH = 1

```

```

# Function to create second modality.
def create_text_FUZZY_2_3(tokenizer, labels):
    text = []

```

```

    for label in labels:
        if label == 0:
            text.append('light')
        if label == 1:
            text.append('light')
        if label == 2:
            text.append('heavy')
        if label == 3:
            text.append('heavy')
        if label == 4:
            text.append('heavy')

```

```

    text = tokenizer.texts_to_sequences(text)
    text = pad_sequences(text)
    return text

```

```

# Create second modality for training and test set.
vocabulary = ['light', 'heavy']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

```

```

train_text = create_text_FUZZY_2_3(tokenizer, train_labels)
valid_text = create_text_FUZZY_2_3(tokenizer, valid_labels)
test_text = create_text_FUZZY_2_3(tokenizer, test_labels)

```

#Modality 2 - FUZZY_3_2 template

```

MAX_WORDS = 2
EMBEDDING_WIDTH = 1

```

```

# Function to create second modality.
def create_text_FUZZY_3_2(tokenizer, labels):
    text = []

```

```

for label in labels:
    if label == 0:
        text.append('light')
    if label == 1:
        text.append('light')
    if label == 2:
        text.append('light')
    if label == 3:
        text.append('heavy')
    if label == 4:
        text.append('heavy')

text = tokenizer.texts_to_sequences(text)
text = pad_sequences(text)
return text

# Create second modality for training and test set.
vocabulary = ['light', 'heavy']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

train_text = create_text_FUZZY_3_2(tokenizer, train_labels)
valid_text = create_text_FUZZY_3_2(tokenizer, valid_labels)
test_text = create_text_FUZZY_3_2(tokenizer, test_labels)

#Modality 2 - FUZZY_4_1 template

MAX_WORDS = 2
EMBEDDING_WIDTH = 1

# Function to create second modality.
def create_text_FUZZY_4_1(tokenizer, labels):
    text = []

    for label in labels:
        if label == 0:
            text.append('healthy')
        if label == 1:
            text.append('healthy')
        if label == 2:
            text.append('healthy')
        if label == 3:
            text.append('healthy')
        if label == 4:
            text.append('ill')

    text = tokenizer.texts_to_sequences(text)
    text = pad_sequences(text)
    return text

# Create second modality for training and test set.
vocabulary = ['healthy', 'ill']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

train_text = create_text_FUZZY_4_1(tokenizer, train_labels)
valid_text = create_text_FUZZY_4_1(tokenizer, valid_labels)
test_text = create_text_FUZZY_4_1(tokenizer, test_labels)

```

ДОДАТОК В

Код реалізації програмного засобу для проведення дослідження впливу мультимодального доповнення метаданих на точність багатокласової класифікації

Лістинг В.1 – Зразок коду для створення текстової модальності

```

MAX_WORDS = 2
EMBEDDING_WIDTH = 1

# Function to create second modality.
def create_text_FUZZY_single_class(tokenizer, labels):
    text = []

    for label in labels:
        if label == int(CLASS_MARK):
            text.append('this')
        else:
            text.append('other')

    text = tokenizer.texts_to_sequences(text)
    text = pad_sequences(text)
    return text

# Create second modality for training and test set.
vocabulary = ['this', 'other']
tokenizer = Tokenizer(num_words=MAX_WORDS)
tokenizer.fit_on_texts(vocabulary)

train_text = create_text_FUZZY_single_class(tokenizer, train_labels)
valid_text = create_text_FUZZY_single_class(tokenizer, valid_labels)
test_text = create_text_FUZZY_single_class(tokenizer, test_labels)

```

Лістинг В.2 – Зразок коду опрацювання та візуалізації отриманих результатів на прикладі набору даних PathMNIST

```

import numpy as np
import os

figures_dir = "./PLOTS"
CLASS_MARK = 'MATH'
LABEL_SCHEME = 'MULTI_' + CLASS_MARK
MODEL = 'ResNet50'
MODEL_NAME = MODEL + '_' + LABEL_SCHEME
SIZE = 32
WEIGHTS = None
MODALITY = 'MULTI'
data_flag = 'pathmnist'
DATASET = data_flag
DA_TYPE = 'NODA'
SAVED_MODEL = 'MAX_AUC'

model_filename_prefix = MODEL_NAME + '_' + str(SIZE) + '_' + str(WEIGHTS) + '_' + MODALITY + '_' + DA_TYPE + '_' + DATASET + '_'

# Single
path_single_std_np = np.array([
    [0.1672,0.0026,0.0397,0.      ,0.0053,0.1034,0.0255,0.      ,0.0441],

```

```

[0.      ,0.0004,0.      ,0.0004,0.      ,0.      ,0.      ,0.      ],
[0.0125,0.0133,0.0872,0.0264,0.      ,0.0629,0.      ,0.0333,0.0172],
[0.0007,0.0006,0.1096,0.0798,0.0006,0.      ,0.0359,0.0184,0.03  ],
[0.043  ,0.0387,0.0025,0.      ,0.0851,0.0008,0.011  ,0.0123,0.0156],
[0.049  ,0.0514,0.1106,0.0346,0.0019,0.0948,0.      ,0.0277,0.0037],
[0.003  ,0.      ,0.0465,0.019  ,0.0197,0.0006,0.1395,0.0068,0.1061],
[0.0073,0.0065,0.1156,0.0058,0.0072,0.0138,0.0035,0.1127,0.0531],
[0.0009,0.003  ,0.11  ,0.0097,0.0189,0.0015,0.0239,0.0154,0.116  ]])
path_single_mean_np = np.array([
[8.2910e-01, 1.8685e-03, 4.0234e-02, 0.0000e+00, 5.1071e-03, 7.9721e-02,
1.7065e-02, 0.0000e+00, 2.6906e-02],
[0.0000e+00, 9.9980e-01, 0.0000e+00, 1.9677e-04, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00],
[8.8496e-03, 1.3274e-02, 7.4779e-01, 1.1799e-02, 0.0000e+00, 7.6205e-02,
0.0000e+00, 1.2045e-01, 2.1632e-02],
[5.2576e-04, 2.6288e-04, 1.1672e-01, 8.0521e-01, 2.6288e-04, 0.0000e+00,
3.4700e-02, 1.3144e-02, 2.9180e-02],
[6.3446e-02, 1.0966e-01, 3.3816e-03, 0.0000e+00, 7.7842e-01, 9.6618e-04,
1.0628e-02, 1.0467e-02, 2.3027e-02],
[3.4628e-02, 5.5743e-02, 1.2866e-01, 1.5484e-02, 1.1261e-03, 5.7461e-01,
0.0000e+00, 1.8750e-01, 2.2523e-03],
[2.4741e-03, 0.0000e+00, 6.4327e-02, 1.8444e-02, 2.1143e-02, 4.4984e-04,
7.2807e-01, 1.0346e-02, 1.5475e-01],
[5.1465e-03, 7.9177e-03, 3.0483e-01, 5.5424e-03, 1.1481e-02, 1.3064e-01,
5.5424e-03, 4.0538e-01, 1.2352e-01],
[8.1103e-04, 1.3517e-03, 9.8270e-02, 1.8248e-02, 1.5139e-02, 6.7586e-04,
4.3796e-02, 1.2301e-02, 8.0941e-01]])

# Select diagonal elements and save them in a list
path_single_mean_diagonal_elements = [path_single_mean_np[i, i] for i in
range(min(path_single_mean_np.shape))]

# Mark = 0
path_0_mean_np = np.array([
[9.9988e-01, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 1.2456e-04,
0.0000e+00, 0.0000e+00, 0.0000e+00],
[0.0000e+00, 1.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
0.0000e+00, 0.0000e+00, 0.0000e+00],
[0.0000e+00, 1.8682e-02, 6.9617e-01, 3.4415e-03, 0.0000e+00, 1.6814e-01,
0.0000e+00, 1.0423e-01, 9.3412e-03],
[0.0000e+00, 0.0000e+00, 3.9958e-02, 8.6856e-01, 0.0000e+00, 0.0000e+00,
5.1788e-02, 3.6803e-03, 3.6015e-02],
[0.0000e+00, 1.1610e-01, 6.2802e-03, 0.0000e+00, 8.1159e-01, 5.1530e-03,
2.8986e-02, 1.0467e-02, 2.1417e-02],
[0.0000e+00, 9.9944e-02, 1.7314e-01, 0.0000e+00, 2.8153e-04, 5.2590e-01,
0.0000e+00, 1.9876e-01, 1.9707e-03],
[0.0000e+00, 0.0000e+00, 5.4206e-02, 3.1489e-02, 2.3167e-02, 1.5744e-03,
7.3054e-01, 7.8722e-03, 1.5115e-01],
[0.0000e+00, 1.1876e-02, 2.7989e-01, 5.9382e-03, 1.0689e-02, 1.4925e-01,
5.1465e-03, 3.9588e-01, 1.4133e-01],
[0.0000e+00, 2.7034e-04, 4.3255e-02, 2.5142e-02, 5.1365e-03, 1.2165e-03,
4.4336e-02, 2.9738e-03, 8.7767e-01]])

# Select diagonal elements and save them in a list
path_0_mean_diagonal_elements = [path_0_mean_np[i, i] for i in
range(min(path_0_mean_np.shape))]
# Perform element-wise subtraction
diff_0_single = [x - y for x, y in zip(path_0_mean_diagonal_elements,
path_single_mean_diagonal_elements)]
k = 0
# Create a new list with the k-th element excluded
new_diff_0_single = diff_0_single[:k] + diff_0_single[k+1:]
# Extract values from the k-th column and save them in a list
k = 0

```

```

error_0_values = path_single_mean_np[:, k].tolist()
# Create a new list with the k-th element excluded
new_error_0_values = error_0_values[:k] + error_0_values[k+1:]

# Plot
import matplotlib.pyplot as plt
from scipy.stats import linregress
import matplotlib.cm as cm

# Generate x values in the range [0, 1]
x = np.linspace(0, .3, 100)
num = 8
# Calculate corresponding y values
y2 = num * x / num
y = x / num

#####
# Averaged by class
#####
Err = [np.mean(new_error_0_values), np.mean(new_error_1_values),
np.mean(new_error_2_values),
        np.mean(new_error_3_values), np.mean(new_error_4_values),
np.mean(new_error_5_values),
        np.mean(new_error_6_values), np.mean(new_error_7_values),
np.mean(new_error_8_values)]
print(Err)
Diff = [np.mean(new_diff_0_single), np.mean(new_diff_1_single),
np.mean(new_diff_2_single),
        np.mean(new_diff_3_single), np.mean(new_diff_4_single),
np.mean(new_diff_5_single),
        np.mean(new_diff_6_single), np.mean(new_diff_7_single),
np.mean(new_diff_8_single)]
print(Diff)
slope, intercept, r_value, p_value, std_err = linregress(Err, Diff)

upper = (slope + std_err) * x + intercept
lower = (slope - std_err) * x + intercept
plt.fill_between(x, lower, upper, color="lavender")

colors = cm.rainbow(np.linspace(0, 1, num + 1))

plt.scatter(new_error_0_values, new_diff_0_single, marker='*', label='$\Delta a(0)$',
color=colors[0], facecolors='none')
plt.scatter(new_error_1_values, new_diff_1_single, marker='s', label='$\Delta a(1)$',
color=colors[1], facecolors='none')
plt.scatter(new_error_2_values, new_diff_2_single, marker='^', label='$\Delta a(2)$',
color=colors[2], facecolors='none')
plt.scatter(new_error_3_values, new_diff_3_single, marker='v', label='$\Delta a(3)$',
color=colors[3], facecolors='none')
plt.scatter(new_error_4_values, new_diff_4_single, marker='p', label='$\Delta a(4)$',
color=colors[4], facecolors='none')
plt.scatter(new_error_5_values, new_diff_5_single, marker='<', label='$\Delta a(5)$',
color=colors[5], facecolors='none')
plt.scatter(new_error_6_values, new_diff_6_single, marker='>', label='$\Delta a(6)$',
color=colors[6], facecolors='none')
plt.scatter(new_error_7_values, new_diff_7_single, marker='o', label='$\Delta a(7)$',
color=colors[7], facecolors='none')
plt.scatter(new_error_8_values, new_diff_8_single, marker='d', label='$\Delta a(8)$',
color=colors[8], facecolors='none')

plt.scatter(np.mean(new_error_0_values), np.mean(new_diff_0_single), marker='*',
label='$\Delta A(0)$', color='black')
plt.scatter(np.mean(new_error_1_values), np.mean(new_diff_1_single), marker='s',
label='$\Delta A(1)$', color='black')

```

```

plt.scatter(np.mean(new_error_2_values), np.mean(new_diff_2_single), marker='^',
label='$\Delta A(2)$', color='black')
plt.scatter(np.mean(new_error_3_values), np.mean(new_diff_3_single), marker='v',
label='$\Delta A(3)$', color='black')
plt.scatter(np.mean(new_error_4_values), np.mean(new_diff_4_single), marker='p',
label='$\Delta A(4)$', color='black')
plt.scatter(np.mean(new_error_5_values), np.mean(new_diff_5_single), marker='<',
label='$\Delta A(5)$', color='black')
plt.scatter(np.mean(new_error_6_values), np.mean(new_diff_6_single), marker='>',
label='$\Delta A(6)$', color='black')
plt.scatter(np.mean(new_error_7_values), np.mean(new_diff_7_single), marker='o',
label='$\Delta A(7)$', color='black')
plt.scatter(np.mean(new_error_8_values), np.mean(new_diff_8_single), marker='d',
label='$\Delta A(8)$', color='black')

# Add labels and a legend
plt.xlabel('$t_{i,k}$, $\sum_i t_{i,k} / (n-1)$')
plt.ylabel('$\Delta a(k)$, $\Delta A(k)$')

#plt.legend(loc='lower right', ncol=2)
plt.legend(loc='upper left', ncol=2)

# Create the first legend for lines 1 and 2
legend1 = plt.legend(loc='upper left', ncol=5, fontsize="8",
labels=['stdev',
'$\Delta a(0)$', '$\Delta a(1)$', '$\Delta a(2)$',
'$\Delta a(3)$', '$\Delta a(4)$',
'$\Delta a(5)$', '$\Delta a(6)$', '$\Delta a(7)$',
'$\Delta a(8)$',
'$\Delta A(0)$', '$\Delta A(1)$', '$\Delta A(2)$',
'$\Delta A(3)$', '$\Delta A(4)$',
'$\Delta A(5)$', '$\Delta A(6)$', '$\Delta A(7)$',
'$\Delta A(8)$'])

plt.plot(x, y, linestyle='--', label='random ($r=1/'+str(num)+'$)', color='b')
plt.text(0.25, 0.01, 'Random ($r=1/'+str(num)+'$)', fontsize=10, ha='center',
color='b')
plt.plot(x, y2, linestyle='dotted', label='best ($r=1$)', color='g')
plt.text(0.13, 0.16, 'Best ($r=1$)', fontsize=10, ha='center', color='g')

# Add the linear regression line to the plot
plt.plot(x, slope * x + intercept, color='black', label='Linear Regression Line')
# Calculate relevant statistics
print(f"Slope (r): {slope}")
print(f"Intercept: {intercept}")
print(f"R-squared: {r_value**2}")
print(f"P-value: {p_value}")
print(f"Standard Error: {std_err}")
# Display statistics in LaTeX format
slope_latex = f'{slope:.3f} $\pm$ {std_err:.3f}'
stats_text = f'Experiment:\nslope ($r$): {slope_latex}\nR-squared:
{r_value**2:.3f}\np-value: {p_value:.3f}'
# Add text label with statistics
plt.text(0.2, -0.15, stats_text, fontsize=10, bbox=dict(facecolor='white', alpha=0.7))

#####
# Assumption:
#####
# If distribution of errors among classes is proportional to the overall accuracy of
the single model,  $A_s$ ,
# then  $\text{slope}=A_s$  with  $R\text{-squared}=1$ .
# Regression Line for Theoretically Assumed values of  $\Delta A(k)$ 
# Path - Accuracy (from scratch): 81.016713%
 $A_s = 0.81016713$ 

```

```

slope = A_s
intercept = 0
# Add the assumed dependence line (see below) to the plot
plt.plot(x, slope * x + intercept, linestyle='--', color='r', label='Theoretically
Assumed Dependence')
plt.text(0.25, 0.135, 'Theory ($r \sim A_s=0.81$)', fontsize=10, ha='center',
color='r')

# Show the plot
plt.grid(True)

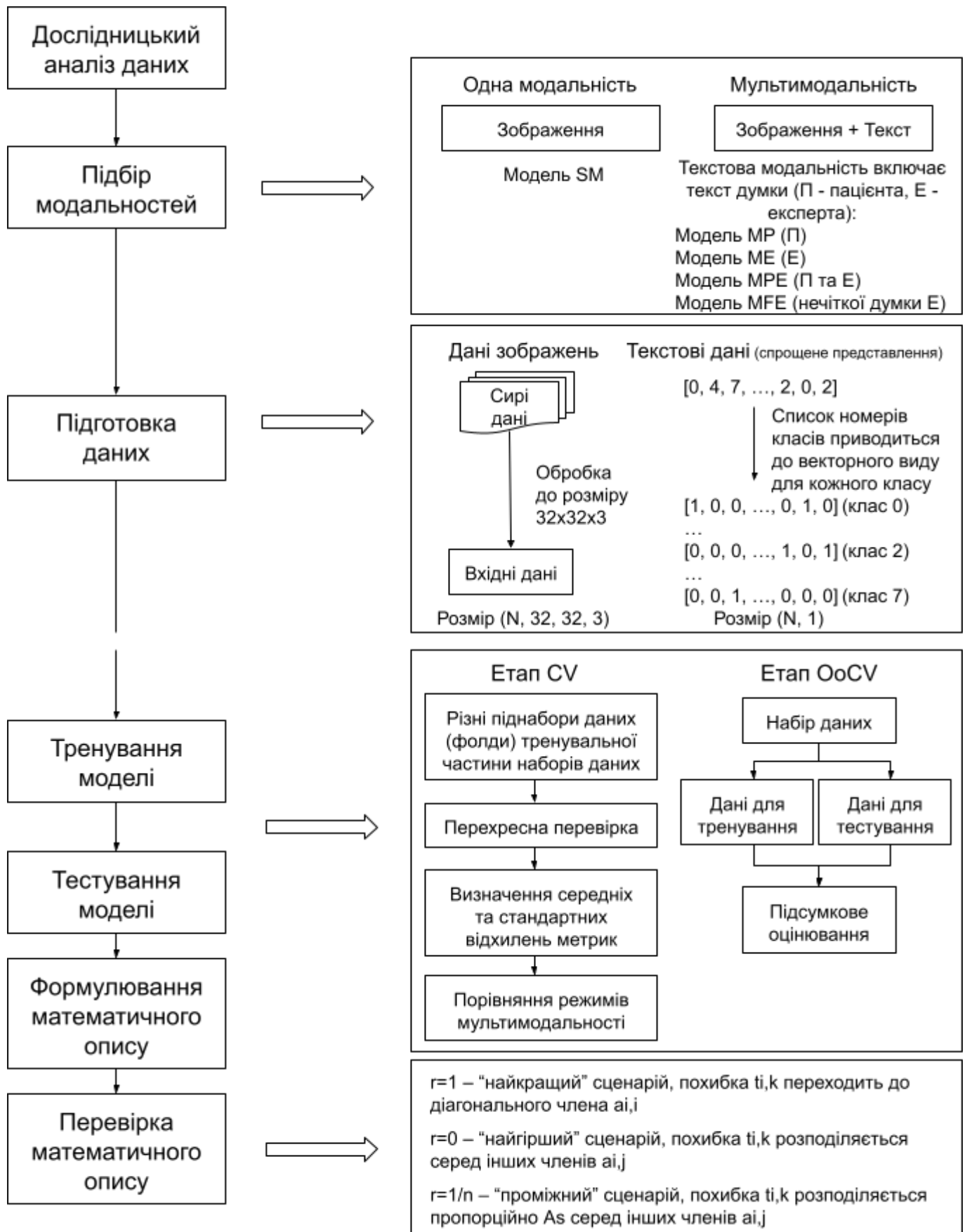
fig_filename = model_filename_prefix + SAVED_MODEL + '.png'
fig_file_path = os.path.join(figures_dir, fig_filename)
plt.savefig(fig_file_path, dpi=300)
fig_filename = model_filename_prefix + SAVED_MODEL + '.eps'
fig_file_path = os.path.join(figures_dir, fig_filename)
plt.savefig(fig_file_path, dpi=300)

plt.show()

```

ДОДАТОК Г

Схема робочого процесу



ДОДАТОК Д

Список публікацій здобувача

1. **Shulha, M.**, Gordienko, Y., Stirenko, S. (2023). Deep Learning with Metadata Augmentation for Classification of Diabetic Retinopathy Level. In: Shakya, S., Balas, V.E., Haoxiang, W. (eds) Lecture Notes in Networks and Systems, vol 587 (pp. 613-630). Springer, Singapore. ISSN 2367-3389. DOI 10.1007/978-981-19-7874-6_46. Scopus, Q4.
2. **Shulha, M.**, Gordienko, Y., Stirenko, S. (2023). Impact of Multimodal Model Complexity on Classification of Diabetic Retinopathy Level. In: García Márquez, F.P., Jamil, A., Eken, S., Hameed, A.A. (eds) Lecture Notes in Networks and Systems, vol 643 (pp. 168-180). Springer, Cham. ISSN 2367-3389. DOI 10.1007/978-3-031-27099-4_13. Scopus, Q4.
3. Gordienko, Y., **Shulha, M.**, Kochura, Y., Rokovyi, O., Alienin, O., Stirenko, S. (2023). Fuzzy Metadata Augmentation for Multimodal Data Classification. In: Shakya, S., Papakostas, G., Kamel, K.A. (eds) Lecture Notes on Data Engineering and Communications Technologies, vol 166 (pp. 157-172). Springer, Singapore. ISSN 2367-4520. DOI 10.1007/978-981-99-0835-6_11. Scopus, Q3.
4. Gordienko, Y., **Shulha, M.**, Kochura, Y., Rokovyi, O., Alienin, O., Taran, V., Stirenko, S. (2023). Ensemble Knowledge Distillation for Edge Intelligence in Medical Applications. In: Pedrycz, W., Chen, SM. (eds) Studies in Computational Intelligence, vol 1100 (pp. 135-168). Springer, Cham. ISSN 1860-9503. DOI 10.1007/978-3-031-32095-8_5. Scopus, Q4.