

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

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

рівень вищої освіти другий (магістерський)

**ДОСЛІДЖЕННЯ ВПЛИВУ МЕТАДАНИХ ДЛЯ**  
**РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ**  
(тема)

Виконав:  
студент 2 курсу, групи ІНФМ-23-2

Маханьов Р.О.  
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Керівник доц. Кобилін О.А.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

Кобилін О.А.  
(прізвище, ініціали)

2025 р.

## Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Маханьову Руслану Олександровичу  
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження впливу метаданих для розпізнавання та класифікації зображень

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 грудня \_\_\_\_\_ 2024 р.3. Вихідні дані до роботи математичні моделі інтеграції метаданих у класифікацію зображень, теоретичні відомості про методи класифікації зображень з використанням метаданих, вибірка зображень для навчання та валідації результатів, перелік використаних програмних засобів: огляд сучасних методів генерації семантичних ознак, механізми уваги та адаптивна комбінація модальностей.

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

1. Огляд методів класифікації зображень з використанням глибоких нейронних мереж.2. Математичні моделі інтеграції метаданих у класифікацію зображень.3. Огляд методів генерації семантичних ознак на основі метаданих.4. Огляд методів роботи з метаданими на основі модальностей.5. Математичні моделі комбінування методів аналізу метаданих.6. Можливості комбінування візуальних ознак та метаданих для покращення точності класифікації.

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

---



---



---



---



---



---



---



---

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

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

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	26.11.24-29.11.24	
3	Аналіз літератури з досліджуваної проблеми	30.11.24-03.12.24	
4	Аналіз технічних засобів	04.12.24-06.12.24	
5	Розробка комп'ютерної моделі	07.12.24-09.12.24	
6	Програмна реалізація	10.12.24-17.12.24	
7	Оформлення пояснювальної записки	18.12.24-20.12.24	
8	Перевірка на плагіат	21.12.2024	
9	Рецензування	22.12.2024	
10	Підготовка презентації та доповіді	23.12.2024	
11	Занесення роботи в електронний архів	01.01.2025	
12	Попередній захист кваліфікаційної роботи	02.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

\_\_\_\_\_ доц. Кобилін О.А.  
(посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 99 с., 1 табл., 66 рис., 1 дод., 46 джерел.

**КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ІНТЕГРАЦІЯ МЕТАДАНИХ, АДАПТИВНА ВАГОВА КОМБІНАЦІЯ МОДАЛЬНОСТЕЙ, ВЗАЄМОДІЯ МІЖ МОДАЛЬНОСТЯМИ, ГЕНЕРАЦІЯ СЕМАНТИЧНИХ ОЗНАК.**

Об'єктом дослідження є методи класифікації зображень з використанням глибоких нейронних мереж.

Метою дослідження є розробка методів інтеграції та аналіз впливу метаданих для розпізнавання та класифікації зображень.

Використано методи математичного моделювання та експериментального тестування. Проведено дослідження різних методів інтеграції метаданих, їхнього впливу на точність класифікації та обчислювальну ефективність моделей. Розроблено та впроваджено алгоритми для адаптивної вагової комбінації модальностей, взаємодії між модальностями, механізму уваги до метаданих та генерації семантичних ознак, а також їх комбіновані варіанти.

У результаті дослідження здійснено програмну реалізацію моделей інтеграції метаданих, що дозволяє значно підвищити точність класифікації зображень у порівнянні з базовими моделями без метаданих.

**CLASSIFICATION OF IMAGES, INTEGRATION OF METADATA, ADAPTIVE WEIGHT COMBINATION OF MODALITIES, CROSS-MODALITY INTERACTION, SEMANTIC FEATURE GENERATION.**

The object of research is image classification methods using deep neural networks.

The aim of the research is to develop methods for integrating and analyzing the impact of metadata for image recognition and classification.

The methods of mathematical modeling and experimental testing were used. Different methods of metadata integration, their impact on classification accuracy and computational efficiency of the models were studied. Algorithms for adaptive weighting of modalities, interaction between modalities, metadata attention mechanism, and semantic feature generation, as well as their combined variants, were developed and implemented.

The study resulted in the software implementation of metadata integration models, which can significantly improve the accuracy of image classification compared to basic models without metadata.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	7
Вступ.....	8
1 Огляд основних методів .....	9
1.1 Визначення метаданих у контексті розпізнаванні зображень.....	9
1.1.1 Основні типи метаданих.....	9
1.1.2 Значення метаданих у контексті обробки зображень .....	11
1.2 Традиційні методи розпізнавання зображень .....	12
1.3 Методи на основі машинного навчання .....	14
1.3.1 Використання метаданих у навчанні моделей .....	15
1.3.2 Мережі для класифікації зображень .....	16
1.3.3 Поєднання метаданих та ознак зображень .....	16
1.4 Моделі обробки метаданих .....	17
1.4.1 Методи обробки метаданих для підвищення якості даних ..	17
1.4.2 Інтеграція метаданих у процес класифікації.....	19
1.5 Проблеми та виклики використання метаданих.....	20
1.5.1 Проблеми стандартизації метаданих .....	20
1.5.2 Неповні або неточні метадані .....	20
1.5.3 Питання конфіденційності та безпеки даних .....	21
1.6 Постановка задачі дослідження.....	22
2 Математичні моделі роботи з метаданими.....	24
2.1 Базова модель без метаданих.....	24
2.2 Інтеграція метаданих у базову модель .....	29
2.3 Покращені методи обробки метаданих .....	31
2.3.1 Адаптивна вагова комбінація модальностей.....	31
2.3.2 Взаємодія між модальностями.....	33
2.3.3 Механізм уваги до метаданих.....	34
2.3.4 Генерація семантичних ознак .....	36
2.4 Комбіновані методи інтеграції метаданих .....	39

	6
2.4.1 Поєднання AWCM та СМІ.....	39
2.4.2 Поєднання AWCM, СМІ та SFG.....	42
3 Дослідження комп'ютерної моделі впливу метаданих .....	45
3.1 Обґрунтування вибору середовища програмної реалізації .....	45
3.1.1 Вибір мови програмування Python.....	45
3.1.2 Використання Google Colab.....	46
3.1.3 Вибір бібліотек .....	47
3.1.4 Вибір набору даних.....	50
3.2 Програмна реалізація.....	51
3.2.1 Аналіз набору даних .....	52
3.2.2 Реалізація базової моделі класифікації зображень .....	54
3.2.3 Додавання метаданих до моделі класифікації зображень ....	58
3.2.4 Реалізація методів покращеної роботи з метаданими .....	60
3.2.5 Реалізація комбінування методів роботи з метаданими .....	62
3.3 Інструкція користувача .....	63
3.4 Тестування розробленої моделі.....	66
Висновки .....	75
Перелік джерел посилання .....	77
Додаток А Результати дослідження .....	83

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

CNN – Convolutional Neural Network (згорткова нейронна мережа)

AWCM – Adaptive Weight Combination of Modalities (адаптивна вагова комбінація модальностей)

CMI – Cross-Modality Interaction (взаємодія між модальностями)

MAM – Metadata Attention Mechanism (механізм уваги до метаданих)

SFG – Semantic Feature Generation (генерація семантичних ознак)

ReLU – Rectified Linear Unit (зрізаний лінійний вузол)

## ВСТУП

Комп'ютерне бачення є однією з найбільш динамічних галузей штучного інтелекту, яка зосереджується на розпізнаванні, класифікації та інтерпретації візуальної інформації зображень та відео. Одним із ключових завдань у цій сфері є класифікація зображень, яка передбачає призначення певного класу або категорії кожному зображенню на основі його візуальних ознак. Це завдання має широке застосування в таких галузях, як медицина, безпека, автомобільна промисловість, системи рекомендацій та управління великими базами даних зображень.

Традиційні методи класифікації, що базуються виключно на візуальних ознаках, часто стикаються з обмеженнями щодо точності та ефективності. Вони не враховують додаткову інформацію, яка може бути представлена метаданими, такими як контекст зйомки, характеристики об'єкта, час зйомки та інші релевантні параметри. Інтеграція метаданих у процес класифікації зображень відкриває нові можливості для підвищення точності моделей, дозволяючи їм використовувати додаткові джерела інформації, що доповнюють візуальні ознаки.

Сучасні дослідження у сфері комп'ютерного бачення активно фокусуються на поєднанні різних типів даних у рамках єдиної моделі, що дозволяє розширити функціональність систем класифікації. Такий підхід передбачає створення методів, які інтегрують досягнення в галузі обробки зображень, аналізу даних і машинного навчання. Це сприяє не лише підвищенню точності класифікації, а й розширює можливості застосування моделей у задачах, де традиційні методи є малоефективними.

Особливу увагу привертає розробка ефективних методів інтеграції метаданих у моделі глибокого навчання для класифікації зображень. Аналіз і порівняння сучасних підходів у цій сфері є важливими для створення високоточних моделей, здатних успішно працювати в різних практичних умовах і забезпечувати оптимальні результати.

# 1 ОГЛЯД ОСНОВНИХ МЕТОДІВ

## 1.1 Визначення метаданих у контексті розпізнаванні зображень

Метадані – це структуровані або напівструктуровані дані, які описують характеристики інших даних. У випадку зображень, метадані надають додаткову інформацію, таку як дата створення, місце зйомки, налаштування камери, та інші атрибути, які не є частиною самого зображення. Вони відіграють критичну роль у процесах обробки та розпізнавання зображень, оскільки можуть допомогти в ефективній класифікації, фільтрації, пошуку і навіть покращенні результатів аналізу зображень. Метадані дозволяють інтегрувати зовнішні контекстуальні фактори, що можуть полегшити ідентифікацію об'єктів або сцен, які важко розпізнати лише на основі візуальної інформації.

### 1.1.1 Основні типи метаданих

Метадані зображень класифікуються за кількома ключовими типами в залежності від того, яку інформацію вони надають і яку роль відіграють у процесах збереження, пошуку та обробки зображень (рис. 1.1).

Технічний тип містить інформацію про технічні характеристики зображення, такі як: формат файлу, розмір файлу, роздільна здатність, колірний простір, інформація про параметри камери та інші. Технічні метадані допомагають зберігати сумісність між різними системами, дозволяють визначати оптимальні налаштування для перегляду або друку зображень, а також полегшують попередню обробку зображень у складних алгоритмах.

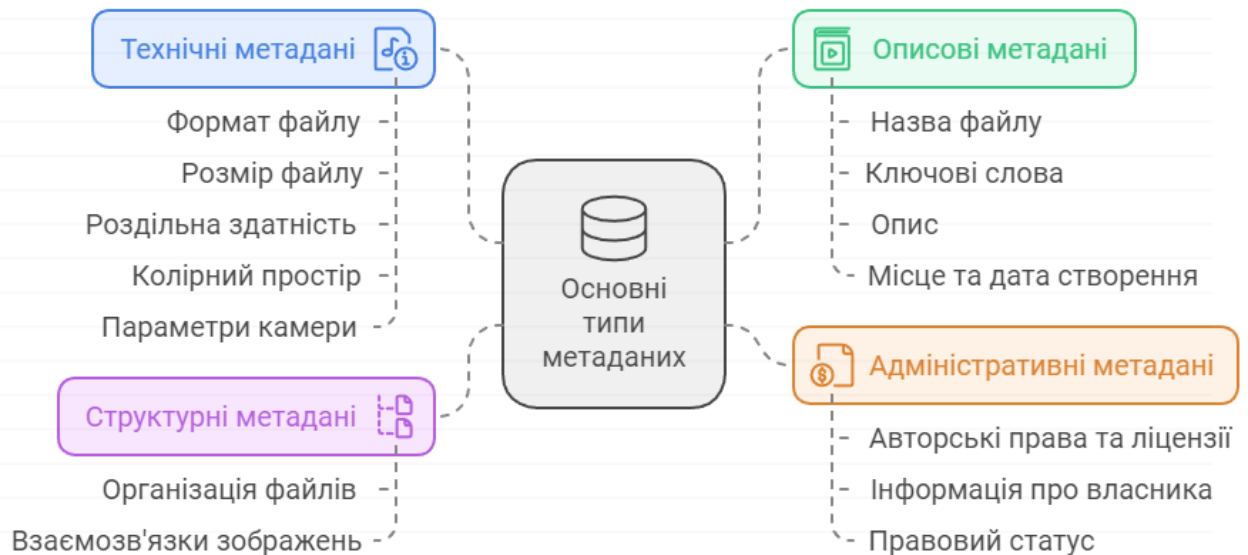


Рисунок 1.1 – Основні типи метаданих

Описові метадані містять інформацію, що описує зміст зображення. Основні це: назва файлу, ключові слова або теги, що відображають основний зміст, опис або коротка анотація зображення, а також місце та дата створення, наприклад, GPS-координати або часові мітки. Описові метадані є основними для процесів пошуку та індексації зображень, оскільки забезпечують семантичний опис того, що відображено на зображенні, дозволяючи швидко знаходити потрібний контент за тематичними запитам.

Адміністративні метадані використовуються для управління ресурсами та включають авторські права та ліцензії на використання, інформацію про власника чи творця файлу та дані про правовий статус зображення, тобто, чи можна його редагувати, розповсюджувати і так далі. Вони допомагають забезпечити законне використання зображень, а також підтримують механізми захисту авторських прав та конфіденційності.

Структурні метадані показують, як окремі частини зображень пов'язані між собою, наприклад, як файли організовані у альбоми, або як кілька зображень утворюють одне панорамне зображення. Вони дозволяють краще організувати та управляти великими колекціями зображень, що важливо в системах, де використовується велика кількість пов'язаних файлів [1–3].

### 1.1.2 Значення метаданих у контексті обробки зображень

Метадані є важливою складовою процесу обробки та розпізнавання зображень, оскільки вони надають інформацію, яка дозволяє алгоритмам краще аналізувати і класифікувати зображення. Основні аспекти їхнього значення показано на рисунку 1.2.



Рисунок 1.2 – Значення метаданих у контексті обробки зображень

Використання метаданих у розпізнаванні зображень суттєво підвищує точність роботи алгоритмів, надаючи їм додаткову інформацію, яка виходить за межі самої візуальної структури зображення. Це допомагає алгоритмам швидше і точніше розпізнавати об'єкти, покращуючи загальну ефективність роботи системи. Уявімо колекцію з мільйонів зображень: замість того, щоб покладатися лише на візуальний аналіз зображень, метадані надають швидкий та точний спосіб фільтрації зображень за змістом або контекстом. Це критично важливо для великих архівів, де візуальні характеристики самі по собі можуть бути недостатньо ефективними для пошуку.

Метадані також спрощують процес автоматичної класифікації зображень, надаючи алгоритмам основний контекст для визначення категорії зображення. Наприклад, метадані можуть вказати на тип зображення, місце його зйомки або використовувану камеру, що дозволяє системі автоматично класифікувати його в певну категорію без необхідності ручного втручання. Це значно підвищує ефективність класифікаційних систем, скорочуючи час на обробку даних та мінімізуючи потребу в ручній роботі.

Також важливо що, метадані сприяють підтримці стандартів та сумісності між різними програмними та апаратними платформами. Вони забезпечують, що зображення можуть зберігатися, передаватися та відтворюватися на різних пристроях без втрати якості або функціональних можливостей. Це особливо важливо у випадках, коли зображення використовуються у міжплатформних середовищах або вимагають специфічних налаштувань для коректного відображення [4–7].

## 1.2 Традиційні методи розпізнавання зображень

Традиційні методи розпізнавання зображень, що розроблялися до ери сучасного машинного навчання та глибокого навчання, засновані на аналізі візуальних характеристик зображення, таких як форми, текстури, кольори та інші геометричні або статистичні ознаки (рис. 1.3). Ці методи зазвичай використовують математичні алгоритми для визначення характеристик зображень і їх подальшої класифікації або пошуку.

Одним із основних підходів у таких методах є аналіз контурів і країв. Алгоритми, такі як оператори Собеля, Канни та Лапласа, аналізують зміни яскравості в зображенні для виявлення його контурів, що дозволяє визначити об'єкти та межі між ними. Інший підхід базується на гістограмі орієнтованих градієнтів (HOG), де векторизується розподіл градієнтів у зображенні, що дозволяє виділяти важливі для розпізнавання об'єкти [8].

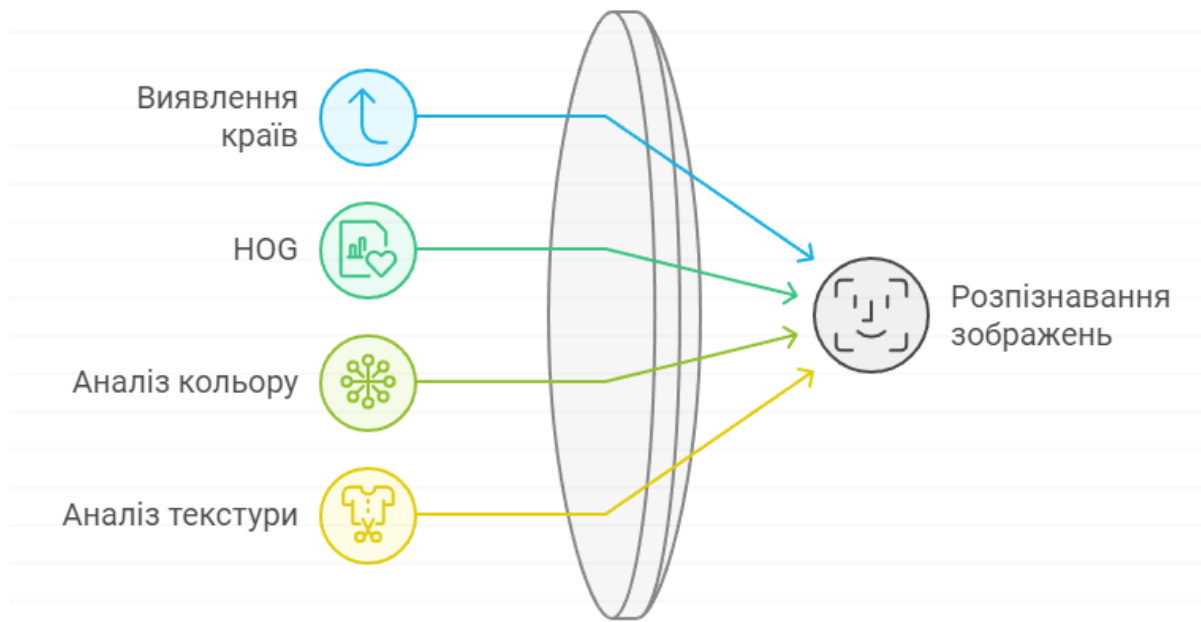


Рисунок 1.3 – Традиційні методи розпізнавання зображень

Ще один популярний метод – аналіз кольорових просторів, де визначаються ключові кольори зображення або його частин. Використання кольорових гістонів дозволяє аналізувати зображення на основі їхньої кольорової композиції, що може бути корисним для класифікації зображень за їх загальними кольоровими характеристиками [9].

Також важливу роль відіграють методи аналізу текстур. Текстура визначає розподіл інтенсивності пікселів у локальних областях зображення, що може бути використано для розпізнавання поверхонь або матеріалів, таких як дерево, метал або тканина. Одним із популярних підходів є використання методів GLCM (матриця співвідношень рівнів сірого), які допомагають визначати текстурні патерни на основі локальних варіацій у зображенні [10].

Традиційні методи, що базуються лише на аналізі візуальних характеристик зображень, мають ряд обмежень. Одне з основних, полягає в тому, що ці методи обмежені у можливостях розпізнавання об'єктів в складних сценах або при змінах умов освітлення, перспективи чи фону. Алгоритми, що покладаються на контури та текстури, можуть втрачати

ефективність, якщо об'єкт частково перекритий або знаходиться у нетипових умовах (наприклад, під різними кутами або у тіні).

Також, такі методи можуть не справлятися з великою різноманітністю зображень. Наприклад, якщо один і той самий об'єкт має різні варіанти текстури або кольору, традиційний підхід не зможе чітко їх класифікувати. Це стає особливо критичним при обробці великих баз даних зображень, де є безліч різних варіантів одного і того ж об'єкта.

Крім того, традиційні методи розпізнавання часто ігнорують семантичний контекст зображення. Алгоритми, що працюють лише з піксельними даними, не здатні розуміти загальний зміст сцени або вловлювати взаємозв'язки між об'єктами на зображенні. Відсутність додаткової інформації у вигляді метаданих може обмежувати алгоритми у здатності розпізнавати об'єкти в контексті.

Ще одне обмеження полягає в тому, що традиційні методи часто є статичними, і їхні моделі не можуть адаптуватися до нових даних або умов без повної переорієнтації. На відміну від сучасних методів на основі машинного навчання, вони не можуть навчатися на нових даних, що робить їх менш гнучкими та менш ефективними в реальних умовах [11–14].

Таким чином, без використання метаданих, традиційні методи значно обмежені у своїй здатності до точної і надійної класифікації зображень в умовах складних або змінюваних обставин.

### 1.3 Методи на основі машинного навчання

Методи розпізнавання та класифікації зображень, засновані на машинному навчанні, є одними з найбільш ефективних і поширених у сучасних системах. Вони використовують алгоритми, що навчаються на великій кількості даних, щоб знаходити приховані закономірності у зображеннях. Основна відмінність цих методів від традиційних полягає в

тому, що вони дозволяють системі самостійно "навчатися" на основі наданих їй прикладів, що дає змогу суттєво підвищити точність і адаптивність при роботі з різноманітними і складними зображеннями [15].

У методах машинного навчання використовуються різні алгоритми, такі як дерева рішень, метод опорних векторів (SVM), та, найголовніше, глибокі нейронні мережі, які зараз є основою багатьох систем для обробки зображень. Однією з ключових переваг цих методів є можливість автоматично виділяти суттєві ознаки зображень без необхідності ручного їх визначення, як це було потрібно в традиційних методах [16].

### 1.3.1 Використання метаданих у навчанні моделей

Метадані можуть бути використані для покращення результатів роботи моделей глибокого навчання. Вони надають додаткову інформацію, що допомагає моделі краще розуміти контекст зображення. Наприклад, якщо модель розпізнає об'єкти на фотографіях, метадані про місце або час зйомки можуть суттєво покращити точність. Це особливо корисно в випадках, коли візуальні ознаки самі по собі не є достатньо інформативними для класифікації.

Використання метаданих у глибокому навчанні може відбуватися шляхом включення їх до вхідних даних моделі разом із піксельними даними зображення. Наприклад, під час навчання можна комбінувати зображення з метаданими про геолокацію, щоб модель змогла краще ідентифікувати об'єкти, типові для певних географічних регіонів. Це підвищує можливості моделі адаптуватися до специфіки зображень і контексту, що покращує загальні результати класифікації. Крім того, аналіз метаданих допомагає виявляти приховані патерни в даних, що можуть бути недоступними при аналізі лише візуальних характеристик, сприяючи розробці більш універсальних і гнучких моделей.

### 1.3.2 Мережі для класифікації зображень

Нейронні мережі, особливо глибокі згорткові нейронні мережі (CNN), є основним інструментом для класифікації зображень у методах машинного навчання. CNN спеціально розроблені для обробки візуальної інформації і здатні автоматично виділяти суттєві ознаки зображень через багаторівневу структуру шарів. На відміну від традиційних методів, де ознаки зображень витягуються вручну, CNN самостійно визначає релевантні ознаки, аналізуючи текстури, форми, кольори та інші аспекти зображення.

Популярними архітектурами нейронних мереж для класифікації зображень є ResNet, VGG, Inception та MobileNet. Ці архітектури здатні обробляти великі обсяги даних і добре справляються з класифікацією навіть у випадках складних і неоднорідних зображень. Вони використовуються в різноманітних сферах – від розпізнавання облич до аналізу медичних зображень [17–19].

### 1.3.3 Поєднання метаданих та ознак зображень

Поєднання метаданих із візуальними ознаками зображень може суттєво підвищити ефективність класифікації. Це дозволяє використовувати як візуальну інформацію, так і контекстуальні дані для точнішого розпізнавання об'єктів. Наприклад, для зображення автомобіля метадані можуть містити інформацію про місце зйомки, що допоможе системі визначити, чи це міський автомобіль чи транспорт, призначений для пересіченої місцевості.

Метадані можуть використовуватися на різних етапах обробки зображень. На етапі передобробки вони можуть допомагати фільтрувати або групувати дані, що дозволяє підвищити ефективність навчання моделі. На етапі навчання метадані можуть бути інтегровані безпосередньо в архітектуру нейронної мережі, додаючись як додатковий вхідний сигнал, що

дозволяє моделі краще інтерпретувати інформацію і приймати рішення на основі ширшого контексту [20, 21].

Зокрема, комбінація візуальних ознак із метаданими може бути використана для поліпшення систем розпізнавання в умовах неповних або неоднозначних зображень. Якщо візуальні ознаки не можуть надати достатньо інформації для точного розпізнавання, метадані можуть додати важливі деталі, які допоможуть моделі ухвалити правильне рішення [22].

#### 1.4 Моделі обробки метаданих

Моделі обробки метаданих відіграють ключову роль у сучасних системах класифікації та розпізнавання зображень, оскільки вони допомагають забезпечити якість даних та їх ефективне використання в процесах машинного навчання. Метадані забезпечують додатковий контекст для зображень, і правильно організовані моделі обробки дозволяють використовувати цю інформацію для покращення результатів класифікації.

Моделі обробки метаданих повинні враховувати різноманітність та неоднорідність самих метаданих, а також забезпечувати їх інтеграцію із візуальною інформацією у спосіб, який підвищує загальну ефективність системи. Це включає методи передобробки, очищення даних, стандартизації та інтеграції в архітектуру класифікації, що забезпечує точність і надійність роботи моделі [23].

##### 1.4.1 Методи обробки метаданих для підвищення якості даних

Обробка метаданих (рис. 1.4) є важливим етапом для забезпечення їхньої якості перед інтеграцією в системи класифікації. Одним із ключових методів обробки є очищення метаданих, яке полягає у виявленні та

виправленні помилок або пропущених даних. Наявність неповних або неправильних метаданих може суттєво знизити точність моделі, тому етап очищення є критичним для забезпечення достовірності інформації.

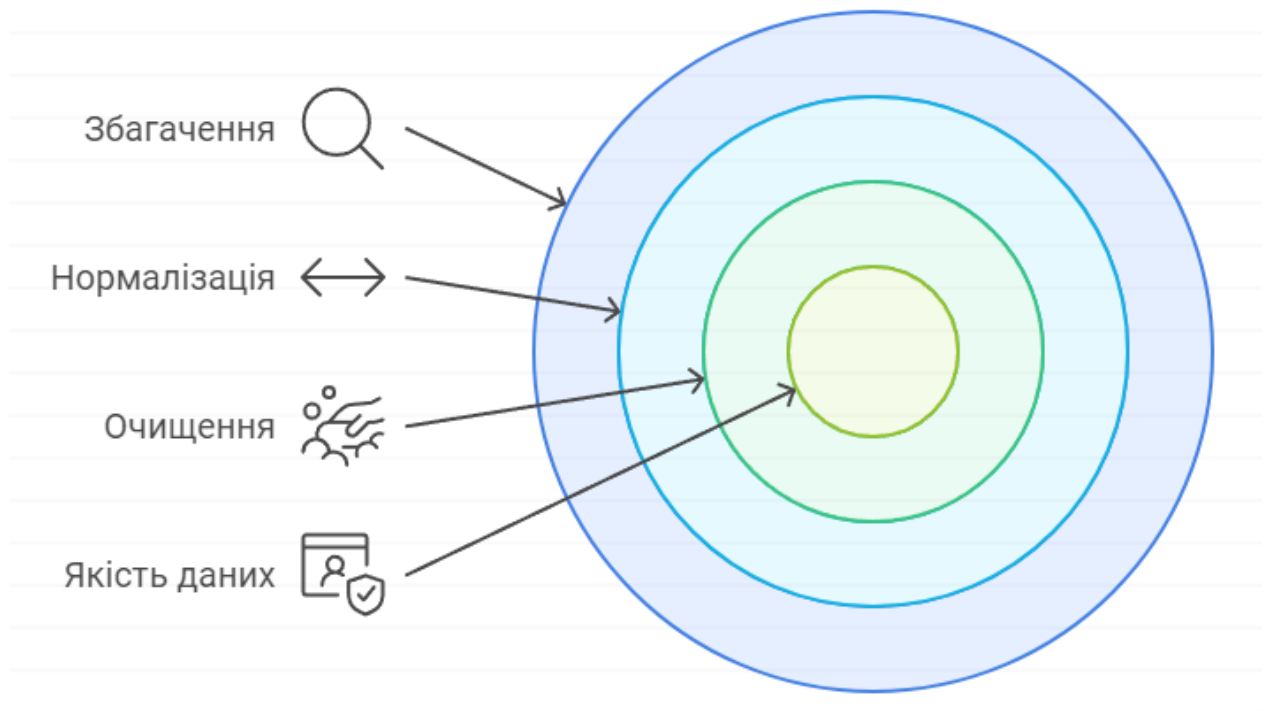


Рисунок 1.4 – Ієрархія обробки метаданих

Іншим важливим методом є нормалізація метаданих. Це процес стандартизації метаданих, при якому всі дані приводяться до єдиного формату або шкали. Наприклад, якщо метадані містять інформацію про дату створення зображення в різних форматах (наприклад, різні системи запису дат), ці дані потрібно стандартизувати для узгодженості та коректної інтерпретації алгоритмами [24, 25].

Також використовуються методи збагачення метаданих, коли система доповнює неповні метадані шляхом збору або інтеграції додаткової інформації з зовнішніх джерел. Це може бути корисним, коли початкові дані містять лише базову інформацію про зображення, але можна додати додаткові атрибути, такі як демографічні дані чи контекст місця зйомки, що покращує якість класифікації.

#### 1.4.2 Інтеграція метаданих у процес класифікації

Сучасні архітектури для класифікації зображень часто інтегрують метадані безпосередньо в процес навчання моделей, що дозволяє використовувати як візуальні ознаки, так і метадані для прийняття рішень. Однією з найпоширеніших підходів є гібридні моделі, де вхідні дані для класифікації складаються не лише з піксельних даних зображення, але й з додаткових даних метаданих, які подаються на інший вхід нейронної мережі. Це дозволяє мережі обробляти як зображення, так і контекстуальні дані паралельно.

Наприклад, мережі з багатовходовою архітектурою дозволяють розглядати зображення та метадані як окремі вхідні сигнали. Один вхід може обробляти піксельну інформацію через згорткові нейронні мережі, тоді як інший вхід приймає метадані. На наступних шарах ці потоки об'єднуються для спільного аналізу і прийняття рішення про класифікацію. Таке об'єднання візуальних ознак та метаданих дозволяє покращити результативність класифікації завдяки врахуванню ширшого контексту зображення [26].

Ще одним підходом є архітектури на основі рекурентних нейронних мереж (RNN), які можуть використовувати метадані для моделювання часових залежностей у класифікації зображень. Наприклад, у випадках, коли важливо враховувати послідовність подій або зміну умов зйомки, метадані можуть надавати важливу інформацію для правильної інтерпретації послідовних зображень.

Також перспективним підходом є використання модулів уваги у нейронних мережах, які дозволяють системам фокусуватися на певних частинах як візуальних даних, так і метаданих. Це дозволяє мережі адаптивно виділяти найбільш релевантну інформацію з метаданих для конкретного завдання, покращуючи точність класифікації. Крім того, модулі уваги забезпечують інтерпретованість роботи моделі.

## 1.5 Проблеми та виклики використання метаданих

Хоча метадані є потужним інструментом для покращення класифікації та розпізнавання зображень, їх використання пов'язане з низкою викликів та проблем. Системи, що покладаються на метадані, можуть стикатися з труднощами, пов'язаними зі стандартизацією, повнотою даних, а також конфіденційністю та безпекою. Ці проблеми можуть суттєво вплинути на ефективність обробки даних та надійність отриманих результатів.

### 1.5.1 Проблеми стандартизації метаданих

Однією з найбільших проблем у використанні метаданих є відсутність єдиних стандартів для їх створення та зберігання. Метадані можуть бути представлені у різних форматах залежно від джерела даних, системи або навіть типу зображення. Це ускладнює їх інтеграцію в загальну систему класифікації [27].

Нестандартність метаданих призводить до проблем з інтерпретацією даних, коли різні системи не можуть коректно обробити та зіставити інформацію через різні формати. Створення і впровадження єдиних стандартів є складним завданням через різноманіття потреб і специфік різних галузей. Це створює ризик неправильного або неповного використання метаданих, що знижує їх ефективність [28].

### 1.5.2 Неповні або неточні метадані

Іншою серйозною проблемою є неповнота або неточність метаданих. Часто зображення супроводжуються неповними метаданими, які можуть не містити ключової інформації, необхідної для коректної класифікації або

пошуку. Наприклад, відсутність даних про місце або час зйомки може ускладнити ідентифікацію контексту зображення.

Крім того, помилкові метадані можуть призводити до хибних результатів класифікації. Наприклад, якщо метадані були згенеровані неправильно або містять помилки це може спричинити неправильну інтерпретацію зображення системою. Виявлення та виправлення таких помилок потребує значних ресурсів, особливо при роботі з великими обсягами даних [29].

Очищення та валідація метаданих є важливим етапом, але це може бути складним процесом, який вимагає автоматичних механізмів виправлення, а також іноді втручання людей для перевірки точності. У системах із великою кількістю зображень помилкові або неповні метадані можуть значно знизити продуктивність та надійність моделі.

### 1.5.3 Питання конфіденційності та безпеки даних

Використання метаданих також пов'язане з серйозними проблемами щодо конфіденційності та безпеки (рис. 1.5). Метадані можуть містити чутливу інформацію, таку як геолокація, дані про автора або інші персональні дані, що можуть використовуватися для ідентифікації особи або місця. У разі витоку або неналежного використання ці дані можуть становити загрозу для конфіденційності та безпеки.

Наприклад, метадані фотографій які містять GPS-координати дозволяють визначити точне місце зйомки. Це може бути використано зловмисниками для відстеження руху людей або для отримання доступу до приватної інформації. Таким чином, без належного управління конфіденційністю та захистом метаданих існує ризик порушення приватності користувачів.

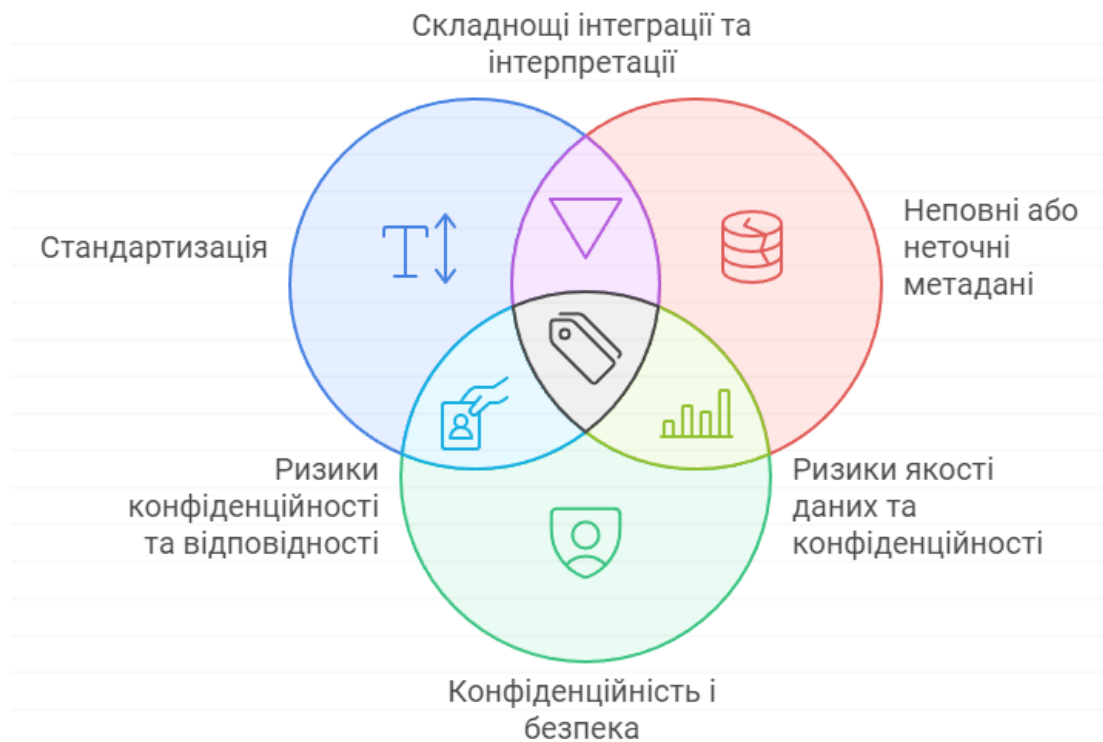


Рисунок 1.5 – Проблеми та виклики використання метаданих

Щоб вирішити ці питання, необхідно впроваджувати механізми захисту метаданих, такі як шифрування та анонімізація чутливої інформації. Крім того, важливо впроваджувати політики, які дозволяють користувачам контролювати, які метадані збираються і як вони використовуються, що забезпечить більшу прозорість і безпеку [30].

Таким чином, метадані, хоч і є потужним інструментом для підвищення точності класифікації, створюють також значні виклики, пов'язані зі стандартизацією, якістю даних та питаннями конфіденційності і безпеки.

## 1.6 Постановка задачі дослідження

Таким чином, використання метаданих для покращення класифікації та розпізнавання зображень є актуальним завданням у сфері обробки візуальної інформації. Метадані, доповнюючи візуальні ознаки, дозволяють підвищити точність класифікації, забезпечують кращий контекст та знижують

залежність від чисто візуальних методів аналізу зображень. У зв'язку з цим постає завдання розробки методів інтеграції та ефективного використання метаданих у процесах класифікації зображень.

Об'єктом дослідження є методи класифікації зображень з використанням глибоких нейронних мереж.

Метою дослідження є розробка методів інтеграції та аналіз впливу метаданих для розпізнавання та класифікації зображень.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих підходів до використання метаданих у класифікації зображень;
- розробити алгоритм інтеграції метаданих у процес класифікації, що поєднує візуальні ознаки та контекстуальні дані;
- реалізувати алгоритми для обробки неповних або неточних метаданих з метою підвищення якості класифікації;
- створити комп'ютерну модель для класифікації зображень із використанням метаданих та перевірити її ефективність на тестових наборах даних.

## 2 МАТЕМАТИЧНІ МОДЕЛІ РОБОТИ З МЕТАДАНИМИ

У цьому розділі розглядаються математичні моделі, розроблені для інтеграції метаданих у процес класифікації зображень. Основою для всіх моделей служить згорткова нейронна мережа (CNN), яка обробляє візуальні ознаки зображень. Для включення метаданих використано різні підходи, кожен з яких описаний математично з відповідними формулами та алгоритмами.

У процесі розробки моделей класифікації зображень з використанням метаданих було обрано комплексний підхід, який дозволяє ефективно інтегрувати додаткову інформацію для підвищення точності класифікації. Основною причиною вибору такого підходу є необхідність покращення продуктивності моделей за рахунок використання не лише візуальних ознак, але й додаткових параметрів, що характеризують об'єкти на зображеннях [31].

### 2.1 Базова модель без метаданих

Базова модель класифікації зображень у даному дослідженні реалізована за допомогою стандартної згорткової нейронної мережі. Ця модель обробляє лише візуальні ознаки зображень без використання додаткових метаданих.

Стандартна CNN складається з кількох послідовних шарів, включаючи згорткові шари, шари підвибірки (пулінгу) та повнозв'язні шари [32].

Згорткові шари відповідають за екстракцію просторових ознак з вхідних зображень. Вони застосовують фільтри (ядра згортки) до вхідних даних, що дозволяє виявляти такі особливості, як краї, текстури та інші важливі характеристики. Після кожного згорткового шару зазвичай слідує

шар активації, наприклад, функція ReLU, яка вводить нелінійність у модель та сприяє кращому навчання.

Шари підвибірки використовуються для зменшення розмірності ознак та зниження обчислювальної складності моделі. Вони допомагають зробити модель стійкішою до змін масштабу та позиції об'єктів на зображенні.

Після кількох згорткових та пулінгових шарів дані передаються у повнозв'язні шари, які виконують класифікацію на основі екстрагованих ознак. Повнозв'язні шари складаються з нейронів, кожен з яких пов'язаний з усіма нейронами попереднього шару. Вихідний шар використовує функцію активації для перетворення вихідних значень у ймовірності для кожного класу [33].

Нехай вхідне зображення позначається як:

$$X \in \mathbb{R}^{H \times W \times C}, \quad (2.1)$$

де  $H$  – висота зображення;

$W$  – ширина зображення;

$C$  – кількість каналів.

Формально, базову модель без використання метаданих можна описати наступним чином:

Крок 1. Опишемо згортковий шар за формулою:

$$Z^{(l)} = \text{ReLU}(W^{(l)} * A^{(l-1)} + b^{(l)}), \quad (2.2)$$

де  $W^{(l)}$  – вагові коефіцієнти  $l$ -го згорткового шару;

$A^{(l-1)}$  – вихід попереднього шару;

$b^{(l)}$  – зміщення  $l$ -го шару;

$*$  – операція згортки;

$\text{ReLU}(x) = \max(0, x)$  – функція активації  $\text{ReLU}$ .

Крок 2. Опишемо шар підвибірки за формулою:

$$P^{(l)} = \text{MaxPool}(Z^{(l)}, k), \quad (2.3)$$

де  $k$  – розмір вікна пулінгу.

Крок 3. Опишемо повнозв'язний шар за формулою:

$$F = \text{ReLU}(W^{(FC)} \cdot P + b^{(FC)}), \quad (2.4)$$

де  $W^{(FC)}$  – вагові коефіцієнти повнозв'язного шару;

$b^{(FC)}$  – зміщення повнозв'язного шару;

• – операція матричного множення.

Крок 4. Опишемо вихідний шар та функцію активації Softmax:

$$y = \text{Softmax}(W^{(out)} \cdot F + b^{(out)}), \quad (2.5)$$

де  $W^{(out)}$  – вагові коефіцієнти вихідного шару;

$b^{(out)}$  – зміщення вихідного шару.

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}. \quad (2.6)$$

Для навчання моделі використовується категоріальна крос-ентропія як функція втрат:

$$\lambda = -\sum_{k=1}^K g_k \log(y_k), \quad (2.7)$$

де  $g_k$  – істинна мітка класу  $k$ ;

$y_k$  – передбачена ймовірність для класу  $k$  ;

$K$  – кількість класів.

Для мінімізації функції втрат застосовується алгоритм оптимізації Adam, який оновлює вагові коефіцієнти  $\theta$  за наступною формулою:

$$\theta = \theta - \eta \cdot \frac{\partial \lambda}{\partial \theta}, \quad (2.8)$$

де  $\eta$  – швидкість навчання.

Схематично базова модель представлена на рисунку 2.1.

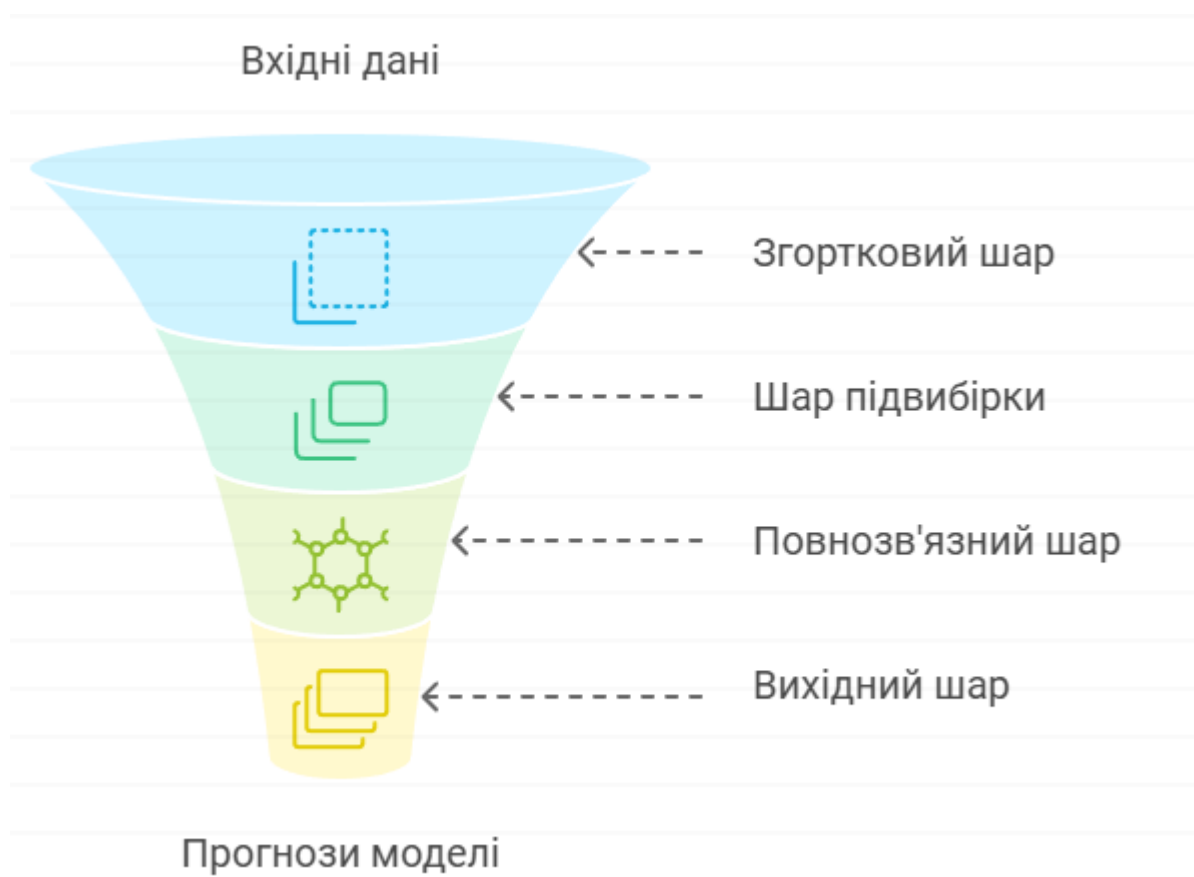


Рисунок 2.1 – Базова модель класифікації зображень

Процедура навчання базової моделі (рис. 2.2) включає наступні кроки:

Крок 1. Ініціалізація ваг: вага  $W$  та зміщення  $b$  ініціалізуються випадковими значеннями.

Крок 2. Прямий прохід: вхідне зображення  $X$  проходить через всі шари мережі, генеруючи передбачення  $y$ .

Крок 3. Обчислення втрат: розраховується значення функції втрат  $\lambda$ .

Крок 4. Зворотний прохід: обчислюються градієнти функції втрат по кожному параметру моделі.

Крок 5. Оновлення ваг: вагові коефіцієнти оновлюються за допомогою алгоритму Adam.

Крок 6. Повторення: процес повторюється протягом заданої кількості епох або до досягнення зупинки за критерієм.

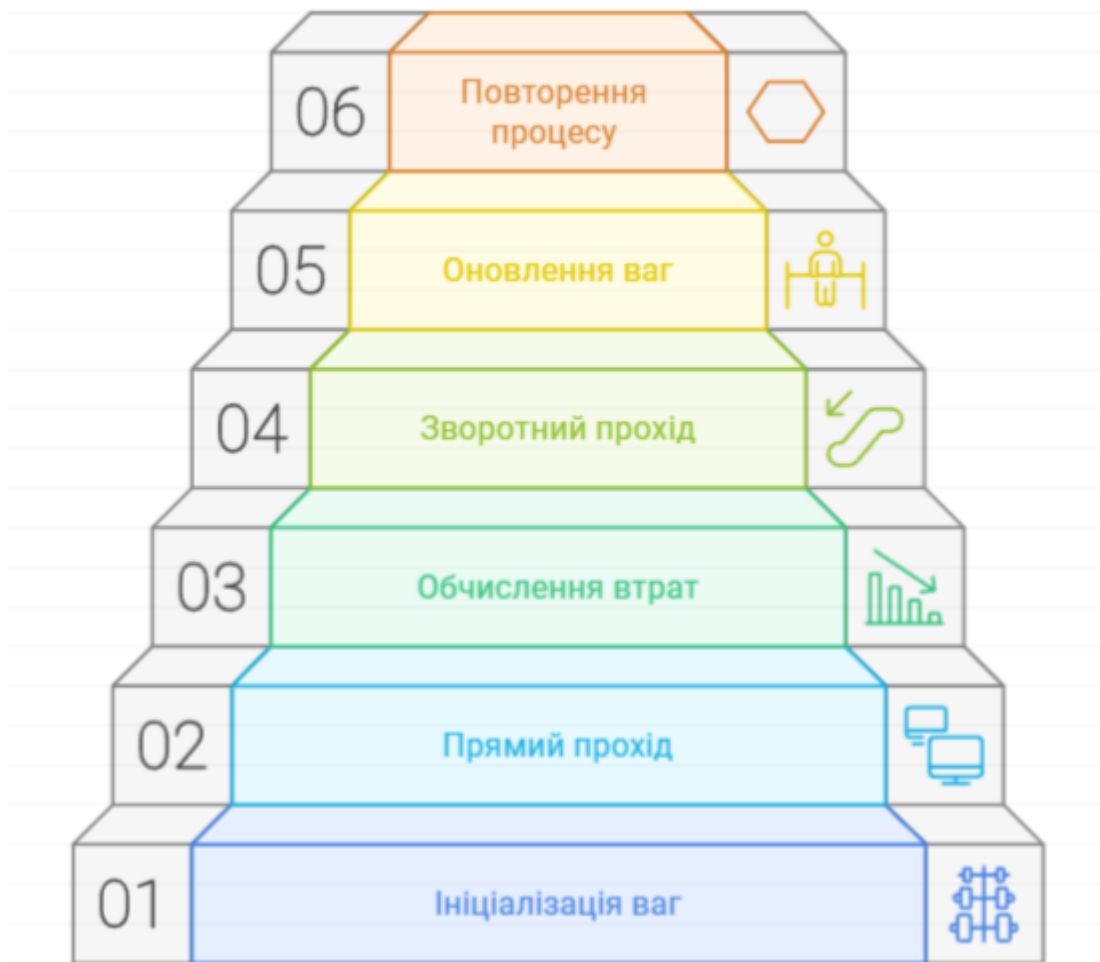


Рисунок 2.2 – Процес навчання нейронної мережі

Базова модель без метаданих демонструє основні принципи роботи згорткових нейронних мереж для класифікації зображень. Вона служить

фундаментом для подальшого розширення моделі шляхом інтеграції метаданих та застосування покращених методів їх обробки, що дозволяє значно підвищити точність класифікації.

## 2.2 Інтеграція метаданих у базову модель

Інтеграція метаданих у базову модель класифікації зображень є ключовим етапом для підвищення її точності та ефективності. Один із найпростіших та найбільш поширених підходів до інтеграції метаданих полягає в об'єднанні ознак шляхом конкатенації векторів візуальних ознак та метаданих. Цей метод дозволяє моделі використовувати як інформацію, отриману з зображень, так і додаткові параметри, що характеризують об'єкти на цих зображеннях [34].

Об'єднання візуальних та контекстуальних ознак для досягнення більш точної класифікації передбачає злиття вихідних даних з CNN та метаданих у єдиний вектор ознак, який потім подається у повнозв'язний шар для подальшої класифікації [35]. Формально цей процес можна описати наступним чином:

Крок 1. Позначимо вихід CNN, що обробляє візуальні ознаки зображення, як:

$$f \in R^F, \quad (2.9)$$

де  $F$  – розмірність вектору ознак з CNN.

Крок 2. Позначимо метадані як:

$$M \in R^D, \quad (2.10)$$

де  $D$  – кількість параметрів метаданих.

Крок 3. Визначемо об'єднаний вектор за формулою:

$$z = [f; M] \in R^{F+D}, \quad (2.11)$$

де ; – операція конкатенації векторів.

Крок 4. Подаємо об'єднаний вектор у повнозв'язний шар за формулою:

$$y = \text{Softmax}(W \cdot z + b), \quad (2.12)$$

де  $W \in R^{K \times (F+D)}$  – матриця ваг повнозв'язного шару;

$b \in R^K$  – вектор зміщення;

*Softmax* – функція активації.

Основні кроки інтеграції метаданих продемонстровані на рисунку 2.3.

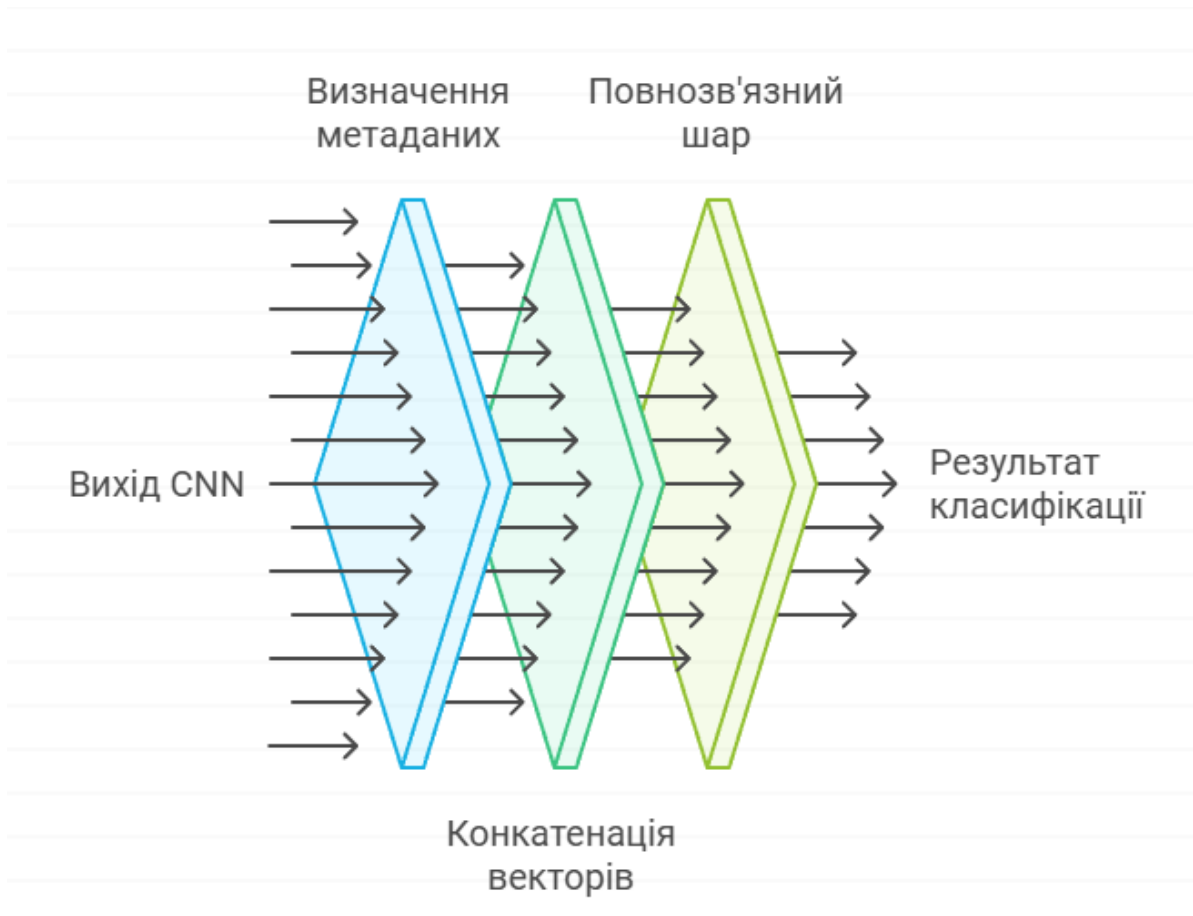


Рисунок 2.3 – Інтеграція метаданих у базову модель

## 2.3 Покращені методи обробки метаданих

Інтеграція метаданих у базову модель класифікації зображень суттєво підвищує її точність та ефективність. Однак, для досягнення максимального ефекту необхідно використовувати більш складні методи обробки метаданих, які дозволяють не лише поєднувати додаткову інформацію з візуальними ознаками, але й оптимізувати цей процес для покращення продуктивності моделі.

### 2.3.1 Адаптивна вагова комбінація модальностей

Метод адаптивної вагової комбінації модальностей передбачає динамічне визначення вагових коефіцієнтів для візуальних ознак та метаданих під час навчання моделі. Основна ідея полягає в тому, щоб дозволити моделі самостійно оптимізувати внесок кожної модальності, забезпечуючи таким чином більш гнучке та ефективне поєднання інформації. Це досягається шляхом навчання вагових коефіцієнтів, які контролюють вплив відповідно візуальних ознак та метаданих на кінцевий вектор ознак (рис. 2.4). Такий підхід дозволяє моделі адаптуватися до різних типів даних та контекстів, підсилюючи ті модальності, які мають більшу релевантність для конкретної задачі класифікації [36–38].

Формально, нехай вихід CNN, що обробляє візуальні ознаки зображення, позначається як  $f \in R^F$ , а метадані представлені вектором  $M \in R^D$ . Об'єднаний вектор ознак  $z$  визначається наступним чином:

$$z = \alpha f + \beta M, \quad (2.13)$$

де  $\alpha$  та  $\beta$  – адаптивні вагові коефіцієнти.

Ці коефіцієнти визначають, наскільки сильно кожна модальність впливає на кінцевий вектор ознак  $z$ . Під час навчання моделі параметри  $\alpha$  та  $\beta$  оптимізуються разом з іншими параметрами мережі за допомогою алгоритму зворотного поширення помилки.



Рисунок 2.4 – Процес адаптивної вагової комбінації

Алгоритмічно процес адаптивної вагової комбінації модальностей можна описати наступним чином:

Крок 1. Екстракція візуальних ознак: вхідне зображення  $X$  обробляється згортковою нейронною мережею, що в результаті генерує вектор ознак  $f$ .

Крок 2. Отримання метаданих: метадані  $M$  представляються у вигляді вектору.

Крок 3. Обчислення ваг: вагові коефіцієнти  $\alpha$  та  $\beta$  ініціалізуються випадковими значеннями та оптимізуються під час навчання нейронної мережі.

Крок 4. Комбінація ознак: вектори  $f$  та  $M$  комбінуються з використанням ваг  $\alpha$  та  $\beta$  для формування вектора  $z$ .

Крок 5. Класифікація: об'єднаний вектор  $z$  подається у повнозв'язний шар для отримання передбачення класу.

Цей метод дозволяє моделі адаптивно регулювати вплив кожної модальності, залежно від їхньої релевантності до поточної задачі класифікації, що сприяє покращенню загальної точності моделі.

### 2.3.2 Взаємодія між модальностями

Взаємодія між модальностями спрямована на встановлення зв'язків між візуальними ознаками та метаданими через мультиплікативні операції (рис. 2.5). Цей метод дозволяє моделі враховувати взаємозалежність даних, що сприяє більш точному розумінню [39].

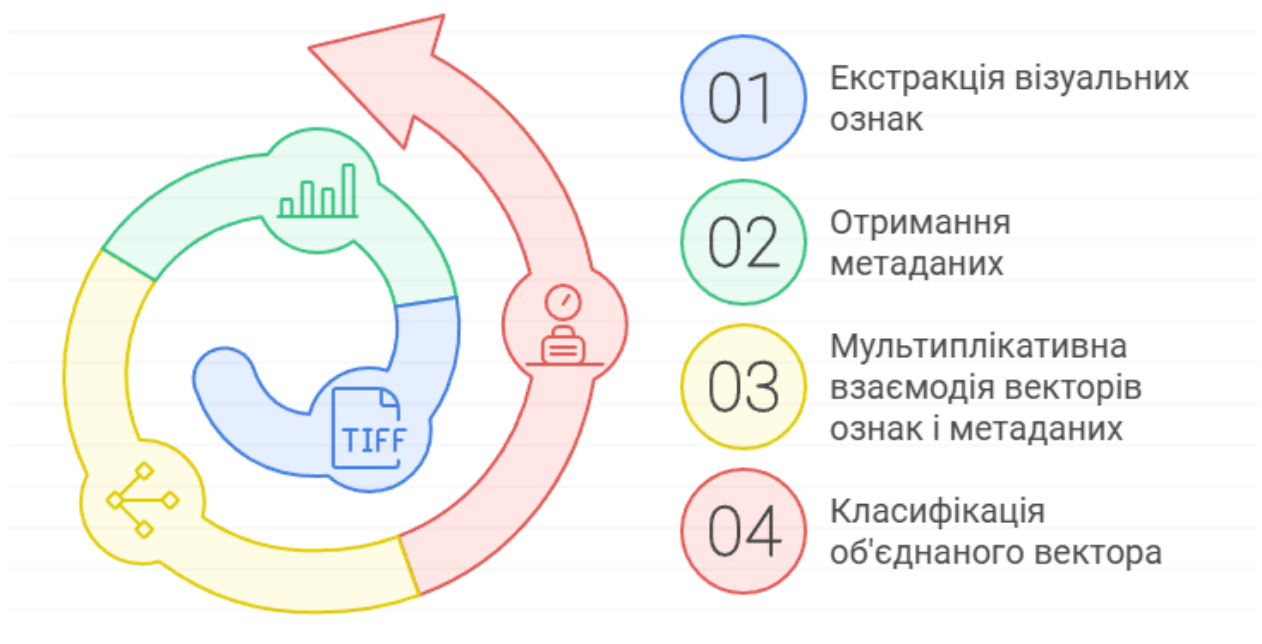


Рисунок 2.5 – Процес взаємодії між модальностями

Початково,  $f$  та  $M$  інтегруються шляхом поелементного множення:

$$z = f \odot M, \quad (2.14)$$

де  $\odot$  – поелементне множення векторів.

Таким чином, кожен елемент вектора метаданих взаємодіє з відповідним елементом візуальних ознак, створюючи нові ознаки, які відображають комбінацію інформації з обох джерел [40]. Цей процес можна описати наступним чином:

Крок 1. Екстракція візуальних ознак: CNN генерує вектор ознак  $f$ .

Крок 2. Отримання метаданих: вектор метаданих  $M$  формується відповідно до обраних параметрів.

Крок 3. Мультиплікативна взаємодія: вектор ознак  $f$  поелементно множиться на вектор метаданих  $M$ , утворюючи новий вектор ознак  $z$ .

Крок 4. Класифікація: об'єднаний вектор  $z$  передається у повнозв'язний шар для класифікації.

Метод взаємодії між модальностями дозволяє моделі враховувати комплексні взаємозв'язки між візуальними ознаками та метаданими, що підвищує здатність моделі до розпізнавання складних патернів у даних. Завдяки цьому підходу модель може ефективніше інтегрувати різномірні джерела інформації, що сприяє покращенню загальної продуктивності та точності класифікації.

### 2.3.3 Механізм уваги до метаданих

Механізм уваги до метаданих спрямований на фокусування моделі на найбільш релевантних частинах метаданих для кожного зображення. Це дозволяє зменшити вплив несуттєвої інформації та підсилити внесок важливих метаданих у процес класифікації [41].

Алгоритм роботи механізму уваги до метаданих включає наступні кроки:

Крок 1. Екстракція візуальних ознак: вектор ознак  $f$  отримується за допомогою CNN.

Крок 2. Отримання метаданих: вектор метаданих  $M$  формується відповідно до вибраних параметрів.

Крок 3. Обчислення ваг уваги: ваги уваги  $A$  обчислюються через функцію  $\text{Softmax}$ , яка застосовується до лінійної трансформації метаданих:

$$A = \text{softmax}(W_a M + b_a), \quad (2.15)$$

де  $W_a \in R^{D \times D}$  та  $b_a \in R^D$  – навчальні параметри механізму уваги.

Ця операція дозволяє моделі визначити, які елементи метаданих є найбільш важливими для конкретного зображення [42].

Крок 4. Модулювання візуальних ознак: вектор ознак  $f$  модулюється за допомогою ваг уваги  $A$ :

$$z = f \odot A, \quad (2.16)$$

де  $\odot$  – поелементне множення векторів.

Таким чином, ваги уваги  $A$  визначають ступінь, до якого кожен елемент метаданих впливає на відповідний елемент візуальних ознак.

Крок 5. Класифікація: об'єднаний вектор  $z$  передається у повнозв'язний шар для отримання кінцевого передбачення класу.

Механізм МАМ дозволяє моделі ефективно зосереджуватися на релевантних метаданих, що підвищує якість класифікації за рахунок підсилення важливих аспектів даних. Завдяки цьому, модель може адаптивно фокусуватися на найбільш значущих елементах метаданих, що мають найбільший вплив на кінцевий результат, покращуючи точність і ефективність класифікації (рис. 2.6).

Такий підхід дозволяє зменшити вплив шумових або незначущих даних, що робить модель більш стійкою та надійною при роботі з великими і різноманітними наборами даних.

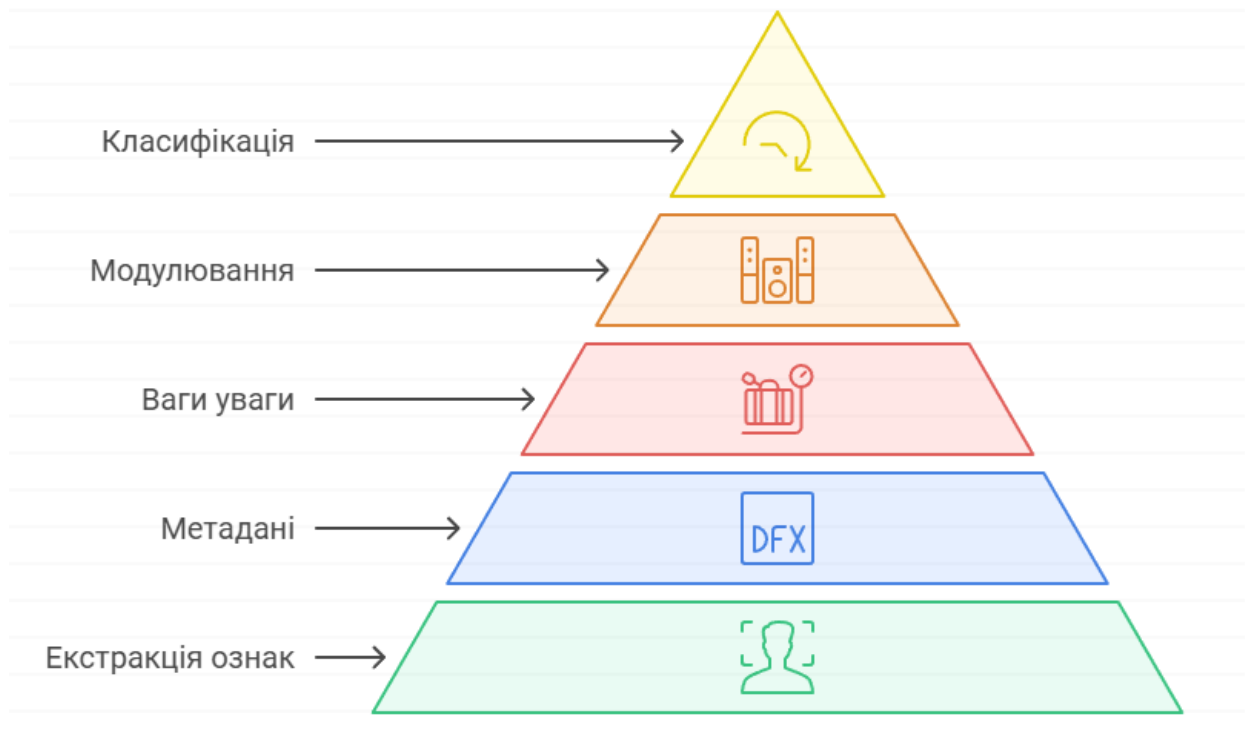


Рисунок 2.6 – Алгоритм механізму уваги до метаданих

#### 2.3.4 Генерація семантичних ознак

Генерація семантичних ознак передбачає створення нових, більш інформативних ознак на основі метаданих за допомогою спеціальної трансформаційної функції. Цей метод дозволяє моделі отримувати глибше розуміння властивостей об'єктів та їхніх взаємозв'язків, що суттєво покращує точність класифікації [43–45].

Процес генерації семантичних ознак включає наступні кроки:

Крок 1. Екстракція візуальних ознак: вектор ознак  $f$  отримується за допомогою CNN.

Крок 2. Отримання метаданих: вектор метаданих  $M$  формується відповідно до вибраних параметрів.

Крок 3. Генерація семантичних ознак: метадані  $M$  передаються через функцію трансформації  $g$ , яка може бути реалізована як повнозв'язний шар

або інша нейронна мережа, для створення нового вектора семантичних ознак  $S$ :

$$S = g(M) \in R^S, \quad (2.17)$$

де  $g$  – трансформаційна функція, що генерує семантичні ознаки з метаданих.

Крок 4. Об'єднання ознак: вектори  $f$  та  $S$  об'єднуються шляхом конкатенації для формування нового вектора ознак  $z$ :

$$z = [f; S] \in R^{F+S}, \quad (2.18)$$

де  $;$  – операція конкатенації векторів.

Крок 5. Класифікація: вектор  $z$  подається у повнозв'язний шар.

Схематично цей процес представлений на рисунку 2.7.

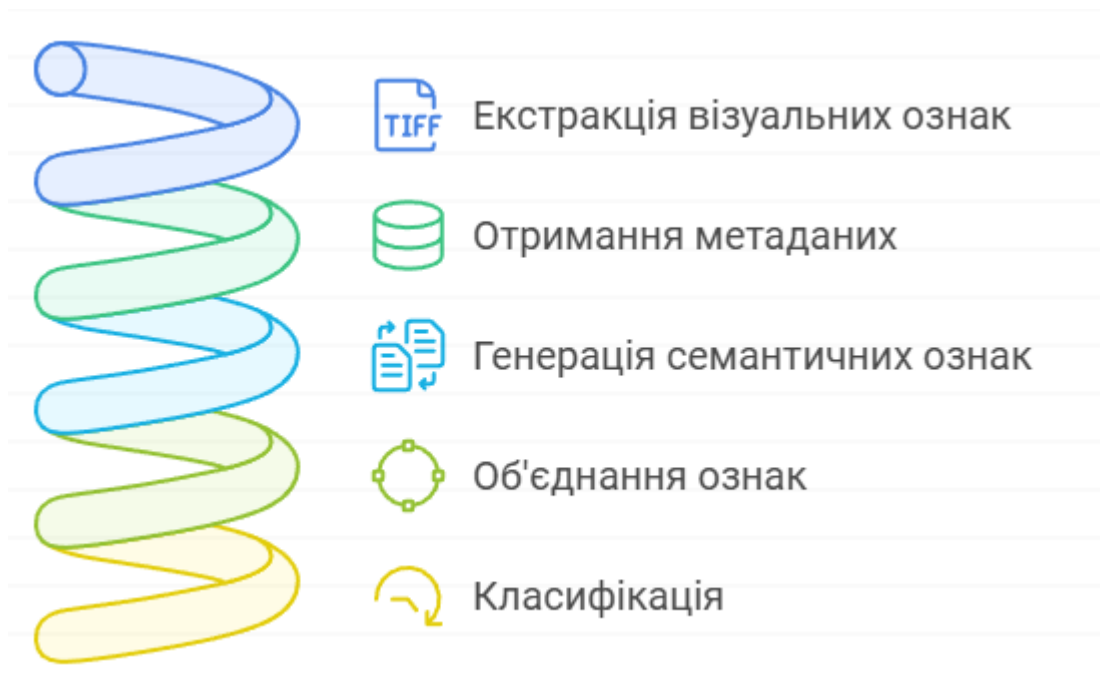


Рисунок 2.7 – Процес генерації семантичних ознак

Алгоритмічно метод генерації семантичних ознак (рис. 2.8) можна описати наступним чином:

Крок 1. Ініціалізація: визначення трансформаційної функції  $g$  для генерації семантичних ознак.

Крок 2. Трансформація метаданих: обчислення  $S = g(M)$ .

Крок 3. Об'єднання ознак: конкатенація  $f$  та  $S$  для створення  $z$ .

Крок 4. Навчання: оптимізація вагових коефіцієнтів у функції трансформації  $g$  та повнозв'язних шарах разом з іншими параметрами моделі за допомогою алгоритму зворотного поширення помилки.

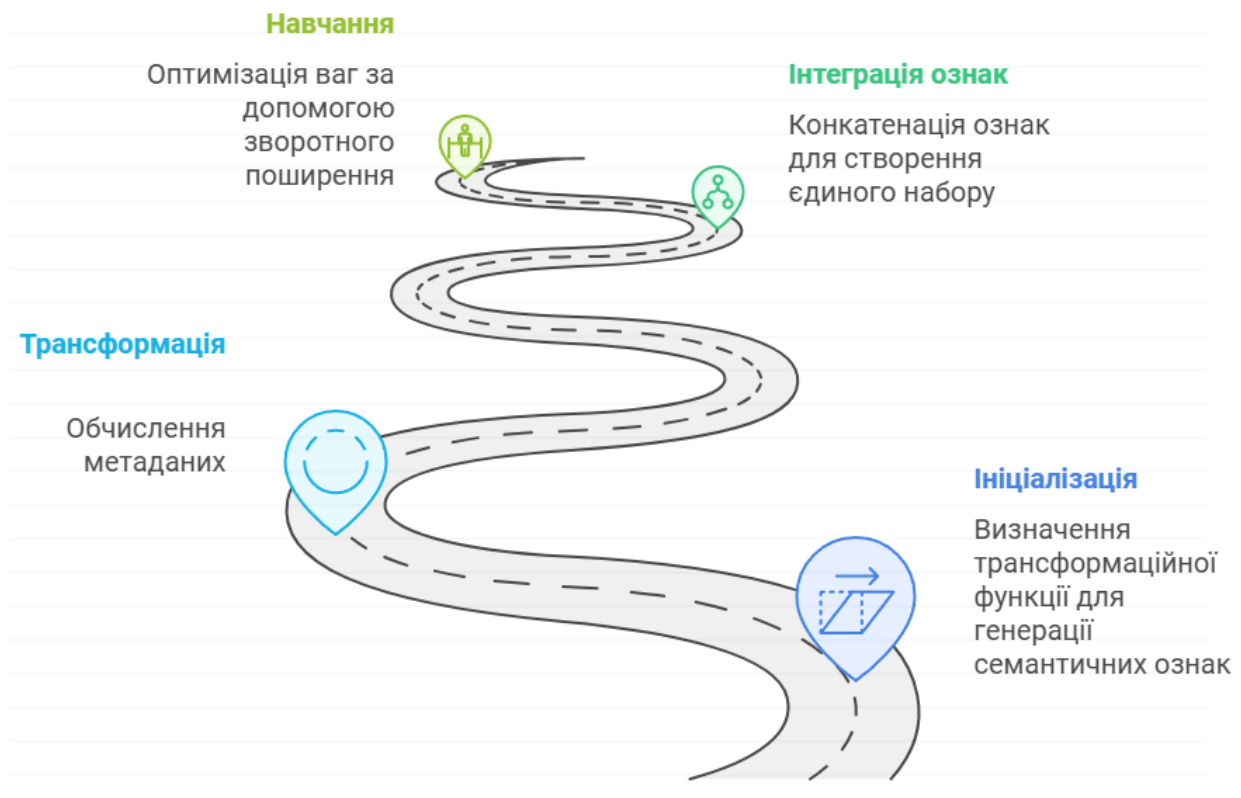


Рисунок 2.8 – Послідовність алгоритму методу генерації семантичних ознак

Метод генерації семантичних ознак дозволяє створювати нові ознаки, які містять більш глибокі семантичні інформації, що покращує здатність моделі до розпізнавання складних патернів у даних.

Цей підхід забезпечує більш комплексне представлення інформації, дозволяючи моделі краще розуміти взаємозв'язки між різними характеристиками об'єктів на зображеннях. Крім того, генерація

семантичних ознак сприяє підвищенню точності класифікації та адаптивності моделі до різноманітних умов.

## 2.4 Комбіновані методи інтеграції метаданих

Комбінування покращених методів інтеграції метаданих дозволяє максимально використовувати переваги кожного з них для підвищення точності класифікації зображень. Це забезпечує більш гнучке та адаптивне використання інформації, що сприяє досягненню кращих результатів у різних сценаріях класифікації.

### 2.4.1 Поєднання AWCM та СМІ

Адаптивна вагова комбінація модальностей та механізм взаємодії між модальностями можуть бути поєднані для досягнення більш гнучкої та інтегрованої моделі. Цей підхід дозволяє моделі не лише адаптивно визначати ваги для кожної модальності, але й встановлювати взаємозв'язки між візуальними ознаками та метаданими через поелементне множення. Завдяки цьому, модель здатна враховувати важливість кожної модальності в конкретному контексті, що дозволяє більш точно і ефективно обробляти складні та багатогранні дані (рис. 2.9).

Математично це можна сформулювати як:

Крок 1. Нехай  $f \in R^F$  – вектор візуальних ознак, отриманий з CNN,  $M \in R^D$  – вектор метаданих,  $\alpha, \beta \in R$  – адаптивні вагові коефіцієнти.

Крок 2. Застосуємо адаптивну вагову комбінацію модельностей:

$$z_{AWCM} = \alpha f + \beta M . \quad (2.19)$$

Крок 3. Інтегруємо механізм взаємодії між модальностями:

$$z_{CMI} = f \odot M . \quad (2.20)$$

Крок 4. Визначемо об'єднаний вектор ознак, як сумарний внесок обох методів:

$$z = z_{AWCM} + z_{CMI} = \alpha f + \beta M + f \odot M . \quad (2.21)$$



Рисунок 2.9 – Інтеграція модальностей для підвищення гнучкості моделі

Алгоритмічно комбінування адаптивної вагової комбінації модальностей та механізму взаємодії між модальностями (рис. 2.10) можна описати наступним чином:

Крок 1. Екстракція візуальних ознак: вхідне зображення  $X$  обробляється згортковою нейронною мережею, що в результаті генерує вектор ознак  $f$ .

Крок 2. Отримання метаданих: метадані  $M$  формуються відповідно до обраних параметрів.

Крок 3. Застосування адаптивної вагової комбінації модальностей:

$$z_{AWCM} = \alpha f + \beta M . \quad (2.22)$$

Крок 4. Застосування взаємодії між модальностями:

$$z_{CM} = f \odot M . \quad (2.23)$$

Крок 5. Об'єднання ознак:

$$z = \alpha f + \beta M + f \odot M . \quad (2.24)$$

Крок 6. Класифікація: об'єднаний вектор  $z$  подається у повнозв'язний шар для отримання передбачення класу:

$$y = \text{Softmax}(W \cdot z + b) . \quad (2.25)$$

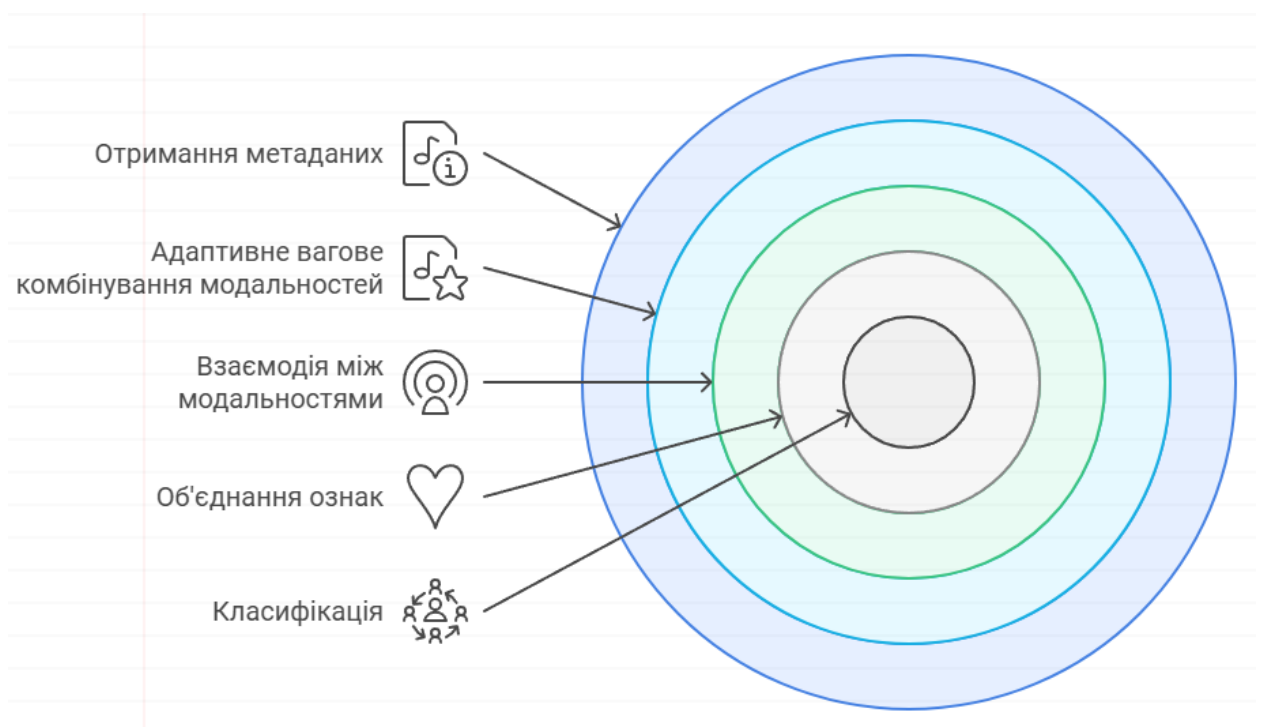


Рисунок 2.10 – Обробка модальностей

Поєднання адаптивної вагової комбінації модальностей та механізму взаємодії між модальностями дозволяє моделі адаптивно визначати ваги для кожної модальності та одночасно враховувати взаємозв'язки між візуальними ознаками та метаданими. Це сприяє більш глибокому інтегруванню інформації та покращує здатність моделі до розпізнавання складних патернів у даних.

#### 2.4.2 Поєднання AWCM, CMI та SFG

Додавання генерації семантичних ознак до адаптивної вагової комбінації модальностей та механізму взаємодії дозволяє моделі не лише адаптивно комбінувати модальності, але й створювати нові, більш інформативні ознаки на основі метаданих, що покращує розуміння властивостей об'єктів і їхніх взаємозв'язків.

Математично це можна сформулювати як:

Крок 1. Нехай  $f \in R^F$  – вектор візуальних ознак, отриманий з CNN,  $M \in R^D$  – вектор метаданих,  $\alpha, \beta \in R$  – адаптивні вагові коефіцієнти,  $S \in R^S$  – вектор семантичних ознак, генерованих з метаданих за допомогою функції  $g$ .

Крок 2. Застосуємо AWCM та CMI.

Крок 3. Згенеруємо семантичні ознаки:

$$S = g(M) \in R^S. \quad (2.26)$$

Крок 4. Визначемо об'єднаний вектор ознак, як сумарний внесок усіх трьох методів:

$$z = \alpha f + \beta M + f \odot M + S. \quad (2.27)$$

Інтеграцію покращених методів до моделі показано на рисунку 2.11.



Рисунок 2.11 – Інтеграція багатофункціональних ознак

Алгоритмічно комбінування адаптивної вагової комбінації модальностей, механізму взаємодії між модальностями та генерації семантичних ознак (рис. 2.12) можна описати наступним чином:

Крок 1. Екстракція візуальних ознак: вхідне зображення  $X$  обробляється згортковою нейронною мережею, що в результаті генерує вектор ознак  $f$ .

Крок 2. Отримання метаданих: метадані  $M$  формуються відповідно до обраних параметрів.

Крок 3. Застосування адаптивної вагової комбінації модальностей та механізму взаємодії між модальностями.

Крок 4. Генерація семантичних ознак.

Крок 5. Об'єднання ознак:

$$z = \alpha f + \beta M + f \odot M + S. \quad (2.28)$$

Крок 6. Класифікація: об'єднаний вектор  $z$  подається у повнозв'язний шар для отримання передбачення класу.

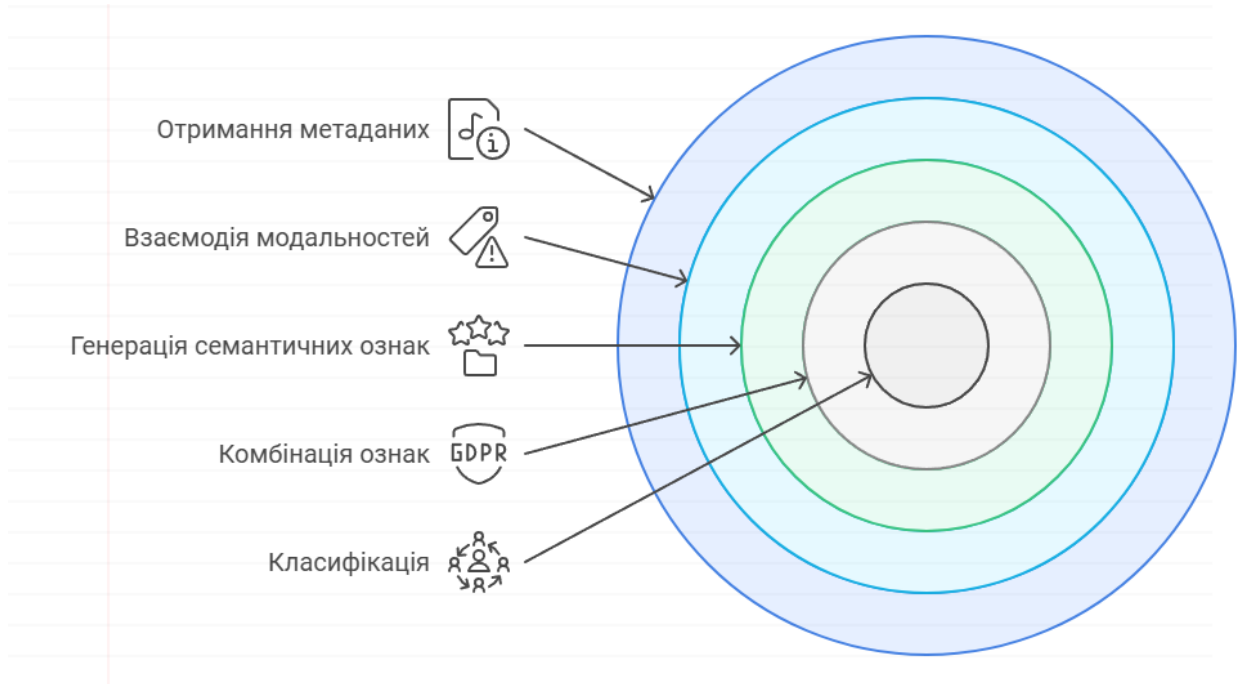


Рисунок 2.12 – Комбінація обробки модальностей та генерації семантичних ознак

### 3 ДОСЛІДЖЕННЯ КОМП'ЮТЕРНОЇ МОДЕЛІ ВПЛИВУ МЕТАДАНИХ

#### 3.1 Обґрунтування вибору середовища програмної реалізації

##### 3.1.1 Вибір мови програмування Python

Вибір Python зумовлений низкою об'єктивних причин. По-перше, Python є високорівневою мовою програмування з простим та інтуїтивно зрозумілим синтаксисом, що значно спрощує процес розробки та зменшує кількість можливих помилок при написанні коду. Це дозволяє зосередитися безпосередньо на вирішенні дослідницьких задач, а не на технічних аспектах програмування.

По-друге, Python має розвинену екосистему бібліотек та фреймворків, спеціально призначених для задач машинного навчання та обробки даних. Зокрема, бібліотеки TensorFlow, TensorFlow Datasets, NumPy, Matplotlib та інші надають потужні інструменти для побудови, тренування та оцінки моделей глибокого навчання, а також для візуалізації та аналізу даних. Ці бібліотеки активно підтримуються спільнотою та мають детальну документацію, що сприяє ефективному їх використанню.

По-третє, Python широко використовується в наукових та індустріальних проектах, пов'язаних з машинним навчанням, що робить його де-факто стандартом у цій галузі. Це забезпечує доступ до великої кількості навчальних матеріалів, прикладів реалізацій та підтримки від спільноти розробників та дослідників. Такий рівень підтримки є критично важливим при вирішенні складних задач та впровадженні новітніх методів.

Крім того, Python підтримує мультипарадигмальне програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне, що надає гнучкість у виборі підходу до реалізації алгоритмів. Це особливо важливо при експериментуванні з різними моделями та методами обробки даних.

Також слід відзначити, що Python є платформонезалежною мовою, що дозволяє запускати розроблені програми на різних операційних системах без потреби в значних змінах коду. У поєднанні з хмарними сервісами, такими як Google Colab, це надає можливість використовувати потужні обчислювальні ресурси без прив'язки до конкретного апаратного забезпечення.

Враховуючи всі перераховані фактори, вибір мови програмування Python для реалізації даного дослідження є обґрунтованим та оптимальним. Її використання забезпечило ефективну розробку моделей глибокого навчання, інтеграцію метаданих у процес навчання та дозволило досягти поставлених цілей дослідження.

### 3.1.2 Використання Google Colab

У рамках даного дослідження для реалізації та тестування моделей глибокого навчання було обрано хмарний сервіс Google Colab. Це інтерактивне середовище для написання та виконання програмного коду на Python, яке надає можливість використовувати потужні обчислювальні ресурси безкоштовно. Вибір Google Colab обумовлений кількома ключовими перевагами, що сприяють ефективній реалізації поставлених задач.

По-перше, Google Colab надає безкоштовний доступ до графічних процесорів (GPU) та тензорних процесорних блоків (TPU), які є необхідними для тренування моделей глибокого навчання. Використання GPU значно прискорює процес навчання нейронних мереж, особливо при роботі з великими наборами даних. Це дозволяє проводити експерименти більш оперативно та ефективно використовувати час дослідження.

По-друге, хмарне середовище Google Colab забезпечує платформонезалежність та доступність з будь-якого пристрою з інтернет-з'єднанням. Це усуває потребу в налаштуванні локального обчислювального середовища та встановленні необхідних бібліотек та пакетів. Усі необхідні

інструменти та залежності можуть бути легко встановлені безпосередньо в середовищі Google Colab за допомогою стандартних команд Python. Такий підхід зменшує технічні бар'єри та спрощує процес розробки.

По-третє, Google Colab має інтеграцію з Google Drive, що забезпечує зручне зберігання, доступ та спільне використання файлів та даних. Це дозволяє легко завантажувати та використовувати набори даних, зберігати проміжні результати та моделі, а також ділитися кодом та результатами з іншими дослідниками або керівником роботи. Така інтеграція сприяє організації роботи та підвищує продуктивність.

Варто також зазначити, що Google Colab регулярно оновлюється та підтримує актуальні версії бібліотек та фреймворків, що використовуються у машинному навчанні. Це гарантує сумісність та доступ до новітніх функціональних можливостей таких бібліотек, як TensorFlow, що є критично важливим для проведення сучасних досліджень у галузі глибокого навчання.

У контексті даного дослідження Google Colab став оптимальним вибором для реалізації моделей розпізнавання та класифікації зображень з урахуванням метаданих. Він забезпечив необхідні обчислювальні ресурси, гнучкість та зручність розробки, що дозволило ефективно провести експерименти та досягти поставлених цілей.

### 3.1.3 Вибір бібліотек

Для успішної реалізації поставлених задач дослідження необхідно було обрати відповідні бібліотеки, які б забезпечили ефективну розробку, навчання та оцінку моделей глибокого навчання з урахуванням метаданих. Після ретельного аналізу доступних інструментів було прийнято рішення використовувати бібліотеки TensorFlow, TensorFlow Datasets, NumPy та Matplotlib.

Бібліотека TensorFlow є одним із найпотужніших та найпоширеніших фреймворків для глибокого навчання, розроблених компанією Google. Вона надає можливість створювати складні нейронні мережі з високою продуктивністю, підтримуючи побудову як простих, так і багатошарових архітектур, включаючи рекурентні та згорткові нейронні мережі. Це особливо важливо для задач розпізнавання та класифікації зображень, де складність моделей відіграє ключову роль. TensorFlow забезпечує швидке навчання моделей на великих наборах даних завдяки підтримці паралельних обчислень на графічних процесорах (GPU) та тензорних процесорних блоках (TPU). Крім того, фреймворк дозволяє налаштовувати моделі під конкретні потреби дослідження, експериментувати з різними архітектурами та гіперпараметрами, що надає гнучкість у пошуку оптимальних рішень. Широка база користувачів та детальна документація спрощують процес освоєння та використання TensorFlow, забезпечуючи підтримку та обмін досвідом у спільноті.

TensorFlow Datasets була обрана як інструмент для зручного доступу до стандартних наборів даних, зокрема PASCAL VOC 2007. Використання TensorFlow Datasets забезпечує автоматизоване завантаження та підготовку даних, включаючи розбиття на тренувальні та тестові вибірки, а також попередню обробку. Це значно спрощує процес підготовки даних, зменшуючи кількість рутинного коду та мінімізуючи ймовірність помилок при обробці. Сумісність з TensorFlow дозволяє легко інтегрувати дані в процес навчання моделей, що прискорює розробку та експерименти. Підтримка багатьох популярних наборів даних дає можливість проводити додаткові дослідження або перевіряти узагальнюваність моделей на інших даних без значних зусиль.

Бібліотека NumPy є фундаментальною для наукових обчислень у Python і відіграє важливу роль у роботі з багатовимірними масивами даних. Вона надає структури даних та функції для роботи з масивами та матрицями, що є основою для багатьох алгоритмів машинного навчання. Використання

оптимізованих алгоритмів у NumPy дозволяє виконувати складні обчислення швидше та ефективніше, ніж при використанні стандартних засобів Python. Це сприяє підвищенню продуктивності коду та зменшенню часу виконання операцій, що є критичним при роботі з великими обсягами даних. Крім того, NumPy є основою для багатьох інших бібліотек машинного навчання, включаючи TensorFlow, що забезпечує сумісність та зручність використання у комплексних проектах.

Для візуалізації даних та результатів моделювання була обрана бібліотека Matplotlib. Вона відіграє ключову роль у процесі аналізу та інтерпретації отриманих результатів, дозволяючи створювати різноманітні типи графіків та діаграм. Використання Matplotlib дає можливість відображати результати тренування моделей, такі як графіки точності та втрат, візуалізувати зображення з передбаченими класами та аналізувати розподіл даних. Інтеграція з інтерактивними середовищами, такими як Google Colab, дозволяє оперативно відображати результати безпосередньо під час виконання коду, що сприяє швидкому прийняттю рішень щодо коригування моделі чи параметрів. Можливість детально налаштовувати вигляд графіків, додавати підписи, легенди та змінювати стилі забезпечує більш інформативне та наочне представлення даних.

Вибір зазначених бібліотек був обґрунтований прагненням забезпечити максимально ефективне та продуктивне середовище для дослідження. Вони надають необхідний функціонал для реалізації складних моделей глибокого навчання, ефективної обробки великих обсягів даних та якісної візуалізації результатів. Використання широко відомих та підтримуваних спільнотою бібліотек також сприяє відтворюваності дослідження, полегшує співпрацю з іншими дослідниками та інтеграцію з додатковими інструментами та фреймворками.

Таким чином, обрані бібліотеки стали невід'ємною частиною програмної реалізації дослідження, дозволивши інтегрувати метадані у процес навчання моделей та досягти значущих результатів у розпізнаванні та

класифікації зображень. Вони забезпечили необхідну інфраструктуру для експериментів та аналізу, що є ключовим для успішного виконання поставлених цілей кваліфікаційної роботи.

#### 3.1.4 Вибір набору даних

У процесі дослідження впливу метаданих на розпізнавання та класифікацію зображень критичним аспектом є вибір відповідного набору даних, який би забезпечив достатню кількість інформації для тренування та оцінки моделей. Після ретельного аналізу доступних наборів даних було обрано PASCAL Visual Object Classes Challenge 2007 (PASCAL VOC 2007) як основний набір даних для даного дослідження.

Вибір PASCAL VOC 2007 обумовлений кількома ключовими факторами, що роблять його оптимальним для поставлених задач. По-перше, цей набір даних є одним із найвідоміших та найпоширеніших у галузі комп'ютерного зору та обробки зображень. Він широко використовується дослідниками для оцінки та порівняння моделей розпізнавання об'єктів, що забезпечує можливість зіставлення отриманих результатів з попередніми дослідженнями та встановлення рівня досягнутої точності.

По-друге, PASCAL VOC 2007 містить різноманітний та збалансований набір зображень, що включає 20 категорій об'єктів, таких як люди, тварини, транспортні засоби та побутові предмети. Така різноманітність забезпечує моделі можливість навчатися на широкому спектрі візуальних характеристик, що підвищує її узагальнюваність та стійкість до варіацій у даних. Це особливо важливо для дослідження впливу метаданих, оскільки різноманітність об'єктів дозволяє оцінити, як додаткова інформація впливає на розпізнавання різних класів.

По-третє, набір даних PASCAL VOC 2007 надає не лише зображення, але й детальні анотації, які включають координати обмежувальних рамок

об'єктів, класи об'єктів та іншу супровідну інформацію. Це дозволяє використовувати його для задач детекції, сегментації та класифікації об'єктів, а також для дослідження впливу різних типів метаданих на продуктивність моделі. Наявність анотацій спрощує процес підготовки даних та забезпечує точність при оцінці результатів.

Крім того, PASCAL VOC 2007 доступний через бібліотеку TensorFlow Datasets, що спрощує процес його завантаження та інтеграції в робочий процес. Це дозволяє автоматично завантажувати, розпаковувати та попередньо обробляти дані, забезпечуючи сумісність з TensorFlow та полегшуючи налаштування даних. Це зменшує кількість рутинного коду та мінімізує ризик помилок при обробці даних.

Також важливим фактором є те, що PASCAL VOC 2007 має встановлені тренувальні та тестові вибірки, що дозволяє проводити коректну оцінку моделі та порівнювати результати.

Вибір PASCAL VOC 2007 як основного набору даних для дослідження дозволив зосередитися на основних цілях роботи – аналізі впливу метаданих на процес розпізнавання та класифікації. Різноманітність та якість даних сприяли отриманню достовірних та значущих результатів, а сумісність з використовуваними інструментами забезпечила ефективність реалізації.

Таким чином, вибір набору даних PASCAL VOC 2007 був обґрунтований його відповідністю вимогам дослідження, високою якістю та широким визнанням у науковій спільноті.

### 3.2 Програмна реалізація

У цьому розділі детально описано процес практичної реалізації дослідження впливу метаданих на розпізнавання та класифікацію зображень. Спочатку було завантажено та проведено аналіз набору даних для ознайомлення з його структурою та особливостями. Далі була реалізована

базова модель класифікації без використання метаданих, що слугувала контрольним зразком для подальших експериментів. Після цього поступово додавалися різні метадані до моделі, і кожного разу проводилася повторна класифікація для оцінки їх впливу. На завершальному етапі були впроваджені різноманітні методи з метою покращення роботи з метаданими та підвищення точності та якості класифікації. Такий поетапний підхід дозволив всебічно дослідити роль метаданих у процесі розпізнавання зображень та визначити оптимальні способи їх інтеграції у моделі глибокого навчання.

### 3.2.1 Аналіз набору даних

Першим етапом практичної реалізації дослідження було завантаження та детальний аналіз набору даних PASCAL VOC 2007. Для цього використовувалася бібліотека TensorFlow Datasets, яка забезпечує зручний доступ до стандартних наборів даних у форматі, сумісному з фреймворком TensorFlow.

Набір даних було завантажено без сегментаційних масок, оскільки основна увага приділялася класифікації об'єктів на зображеннях та використанню метаданих, пов'язаних з ними. Після завантаження було отримано інформацію про структуру набору даних, зокрема списки можливих класів об'єктів та їх поз. Це дозволило зрозуміти, з якими категоріями працюватиме модель та які додаткові характеристики об'єктів можна врахувати.

Для глибшого ознайомлення з даними була розроблена функція візуалізації прикладів з набору даних (рис. 3.1). Ця функція відображала зображення разом з обмежувальними рамками об'єктів, їхніми класами та іншими метаданими, такими як поза об'єкта, ознаки того, чи об'єкт частково прихований або є складним для розпізнавання. Такий візуальний аналіз

дозволив оцінити якість даних, розподіл класів та складність зображень. Для того, щоб ознайомитися з різноманітністю зображень та об'єктів, дані було випадково перемішано. Це забезпечило можливість перегляду різних прикладів з набору даних та виявлення можливих аномалій або складних для класифікації випадків. Під час аналізу було звернено увагу на розміри зображень, кількість об'єктів на кожному з них, а також на те, як метадані можуть вплинути на процес навчання моделі.

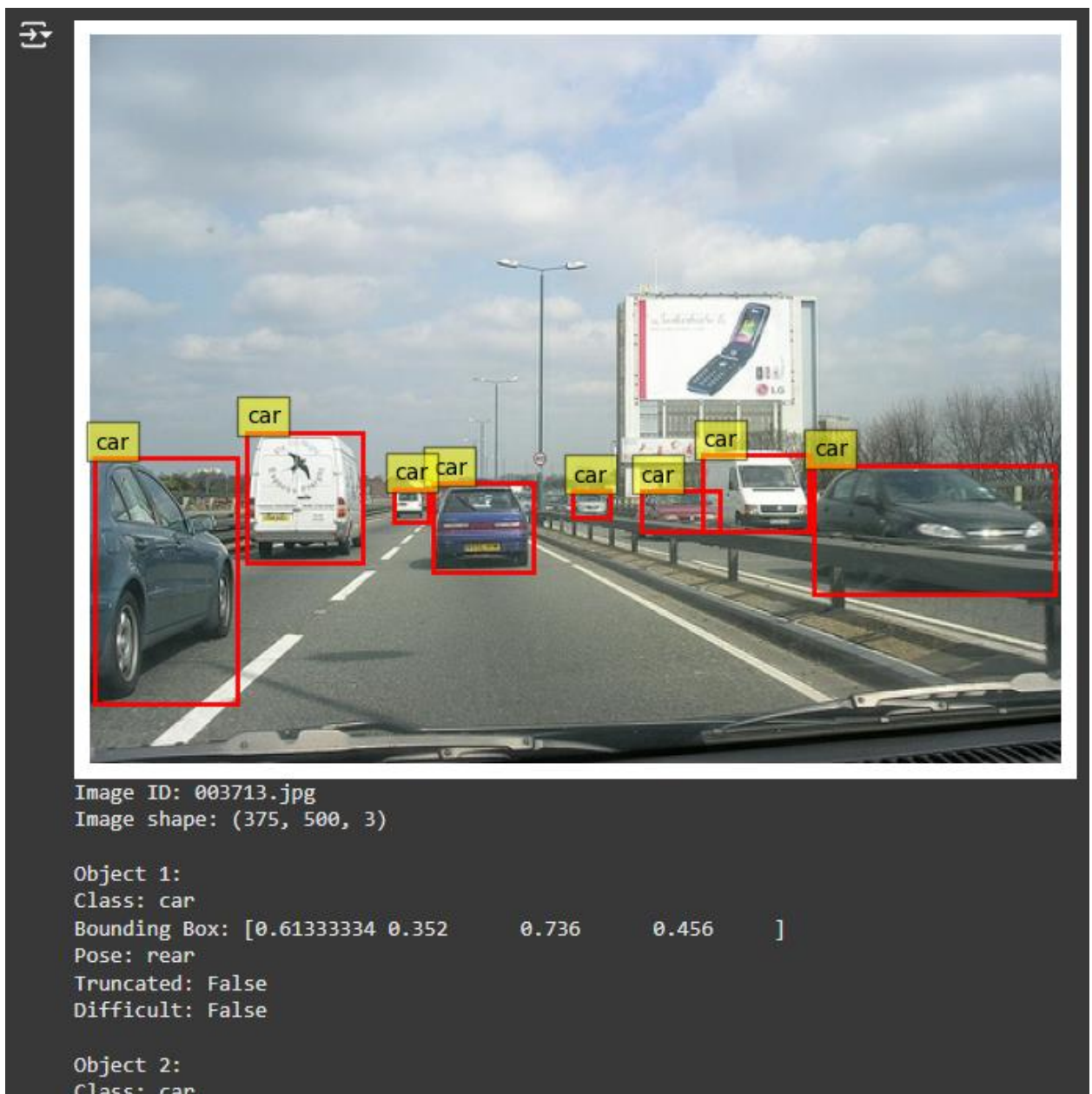


Рисунок 3.1 – Візуалізації прикладів з набору даних

Аналіз показав, що набір даних PASCAL VOC 2007 містить значну кількість зображень з високим рівнем варіативності, як у плані самих об'єктів, так і їхнього розташування, масштабу та поз. Наявність метаданих, таких як пози об'єктів та інформація про складність їх розпізнавання, відкриває можливості для їх використання з метою покращення якості класифікації. Зокрема, було відзначено, що об'єкти з позначкою наявності складних об'єктів можуть впливати на точність моделі та вимагають особливої уваги при навчанні.

### 3.2.2 Реалізація базової моделі класифікації зображень

Після ознайомлення з набором даних та його попереднього аналізу було розпочато розробку базової моделі класифікації зображень без використання метаданих. Ця модель слугувала контрольним зразком для оцінки ефективності подальших покращень та впровадження метаданих у процес навчання.

Для реалізації моделі використовувався фреймворк TensorFlow та його високорівнева API Keras. Базовою архітектурою була обрана попередньо навчена модель ResNet50. Ця модель відома своєю високою точністю у задачах класифікації зображень та глибиною архітектури, що дозволяє ефективно витягувати ознаки з зображень.

На етапі підготовки даних було здійснено завантаження тренувальної та валідаційної вибірок набору даних. Зображення були масштабовані до розміру 224 на 224 пікселів, щоб відповідати вхідним вимогам моделі ResNet50. Також було виконано нормалізацію пікселів шляхом ділення значень на 255, що приводить їх до діапазону від 0 до 1 та сприяє стабільнішому навчанню моделі.

Навчання моделі проводилося на тренувальному наборі даних протягом 20 епох (рис. 3.2), з валідацією на валідаційному наборі після

кожної епохи. Для оцінки ефективності навчання було виміряно час, витрачений на тренування моделі. Це дозволило оцінити продуктивність моделі та можливість її використання у практичних застосуваннях.

```

GPU is available. Training will be done on GPU.
Epoch 1/20
79/79 ----- 56s 507ms/step - accuracy: 0.2282 - loss: 2.7135 - val_accuracy: 0.3394 - val_loss: 2.4294
Epoch 2/20
79/79 ----- 57s 247ms/step - accuracy: 0.3178 - loss: 2.4371 - val_accuracy: 0.3394 - val_loss: 2.3969
Epoch 3/20
79/79 ----- 19s 242ms/step - accuracy: 0.3197 - loss: 2.3984 - val_accuracy: 0.3394 - val_loss: 2.3722
Epoch 4/20
79/79 ----- 17s 198ms/step - accuracy: 0.3200 - loss: 2.3680 - val_accuracy: 0.3394 - val_loss: 2.3530
Epoch 5/20
79/79 ----- 16s 200ms/step - accuracy: 0.3217 - loss: 2.3434 - val_accuracy: 0.3406 - val_loss: 2.3375
Epoch 6/20
79/79 ----- 19s 240ms/step - accuracy: 0.3231 - loss: 2.3228 - val_accuracy: 0.3414 - val_loss: 2.3248
Epoch 7/20
79/79 ----- 19s 213ms/step - accuracy: 0.3248 - loss: 2.3052 - val_accuracy: 0.3414 - val_loss: 2.3141
Epoch 8/20
79/79 ----- 18s 224ms/step - accuracy: 0.3292 - loss: 2.2897 - val_accuracy: 0.3426 - val_loss: 2.3050
Epoch 9/20
79/79 ----- 19s 199ms/step - accuracy: 0.3318 - loss: 2.2758 - val_accuracy: 0.3418 - val_loss: 2.2971
Epoch 10/20
79/79 ----- 20s 194ms/step - accuracy: 0.3322 - loss: 2.2633 - val_accuracy: 0.3414 - val_loss: 2.2901
Epoch 11/20
79/79 ----- 15s 192ms/step - accuracy: 0.3321 - loss: 2.2518 - val_accuracy: 0.3414 - val_loss: 2.2838
Epoch 12/20
79/79 ----- 21s 197ms/step - accuracy: 0.3356 - loss: 2.2412 - val_accuracy: 0.3422 - val_loss: 2.2781
Epoch 13/20
79/79 ----- 16s 199ms/step - accuracy: 0.3366 - loss: 2.2314 - val_accuracy: 0.3430 - val_loss: 2.2728
Epoch 14/20
79/79 ----- 15s 194ms/step - accuracy: 0.3372 - loss: 2.2221 - val_accuracy: 0.3422 - val_loss: 2.2680
Epoch 15/20
79/79 ----- 20s 188ms/step - accuracy: 0.3382 - loss: 2.2133 - val_accuracy: 0.3434 - val_loss: 2.2636
Epoch 16/20
79/79 ----- 21s 189ms/step - accuracy: 0.3400 - loss: 2.2050 - val_accuracy: 0.3438 - val_loss: 2.2594
Epoch 17/20
79/79 ----- 23s 225ms/step - accuracy: 0.3404 - loss: 2.1971 - val_accuracy: 0.3442 - val_loss: 2.2555
Epoch 18/20
79/79 ----- 17s 219ms/step - accuracy: 0.3404 - loss: 2.1895 - val_accuracy: 0.3446 - val_loss: 2.2519
Epoch 19/20
79/79 ----- 22s 235ms/step - accuracy: 0.3445 - loss: 2.1822 - val_accuracy: 0.3446 - val_loss: 2.2484
Epoch 20/20
79/79 ----- 15s 192ms/step - accuracy: 0.3455 - loss: 2.1752 - val_accuracy: 0.3450 - val_loss: 2.2451
Training time: 443.33 seconds
79/79 ----- 7s 92ms/step - accuracy: 0.3337 - loss: 2.2430
Accuracy: 34.50%

```

Рисунок 3.2 – Візуалізація процесу навчання

Оскільки зображення в наборі даних можуть містити кілька об'єктів різних класів, для спрощення задачі класифікації було обрано підхід, де для кожного зображення використовується мітка найвищого класу.

Базова модель складалася з попередньо навченої ResNet50 без верхніх шарів, що дозволяло використовувати її як екстрактор ознак. Шари базової моделі були заморожені, щоб уникнути зміни ваг під час навчання та зосередитися на навчанні верхнього класифікаційного шару. До базової моделі було додано шар глобального усереднення ознак та повнозв'язний шар з кількістю нейронів, що відповідає кількості класів у наборі даних, з

функцією активації softmax для отримання ймовірностей належності до кожного класу.

Модель була скомпільована з оптимізатором Adam та функцією втрат «sparse\_categorical\_crossentropy», яка підходить для задач багатокласової класифікації з цілочисловими мітками класів. Метрикою оцінки було обрано точність, що дозволяє відслідковувати долю правильно класифікованих зображень.

Після завершення навчання модель була протестована на валідаційному наборі даних. Було обчислено значення функції втрат та точність моделі на цьому наборі. Результати показали базовий рівень продуктивності, який слугував відправною точкою для подальших покращень та додавання метаданих.

Для візуальної оцінки роботи моделі були вибрані зображення з валідаційного набору, на яких модель здійснювала передбачення класів. Ці зображення разом з передбаченими класами були відображені (рис. 3.3), що дозволило оцінити, наскільки добре модель розпізнає об'єкти на практиці.

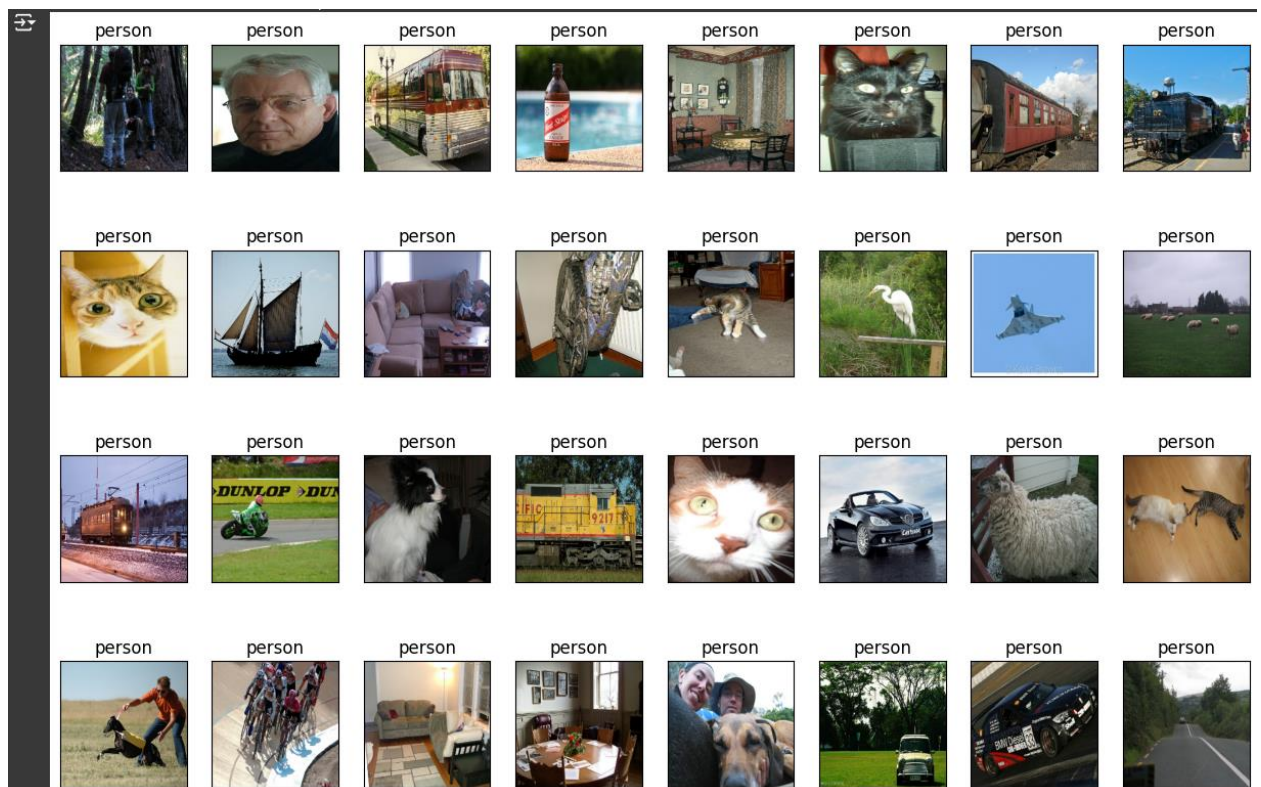


Рисунок 3.3 – Зображення з валідаційного набору

Також були побудовані графіки залежності точності (рис. 3.4) та втрат (рис. 3.5) від кількості епох для тренувального та валідаційного наборів даних. Аналіз цих графіків надав можливість виявити тенденції в навчанні моделі, такі як можливе перенавчання або недонавчання, та зробити висновки щодо необхідності коригування гіперпараметрів або структури моделі.

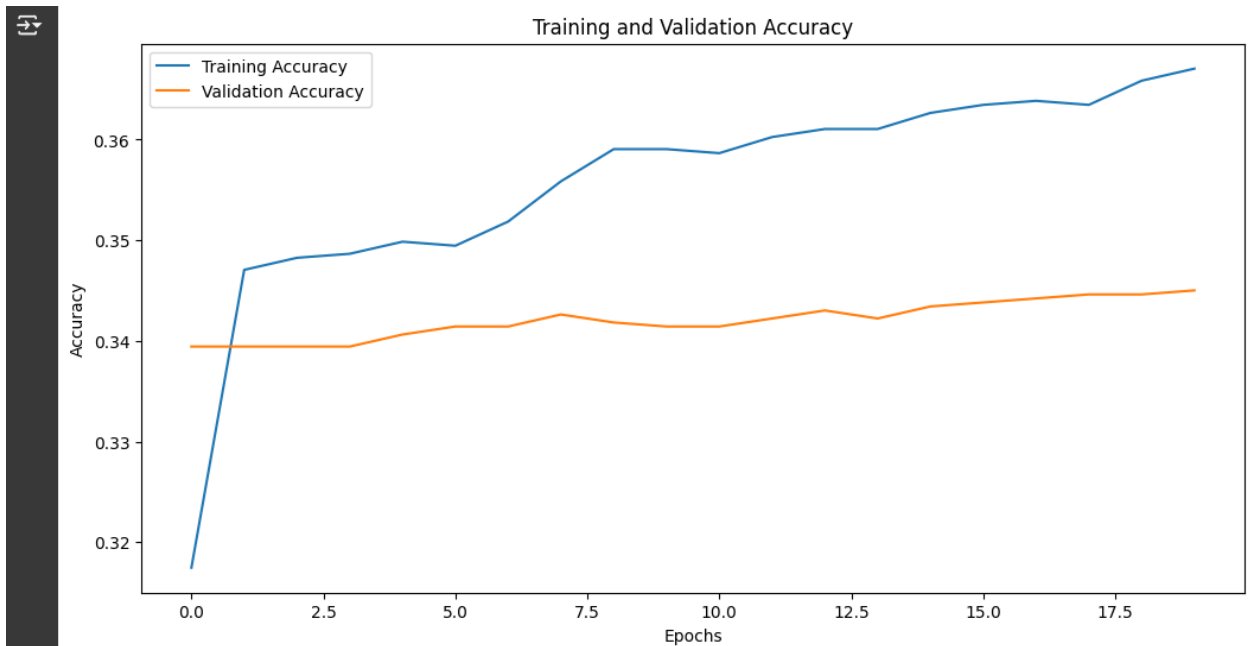


Рисунок 3.4 – Графік залежності точності від кількості епох

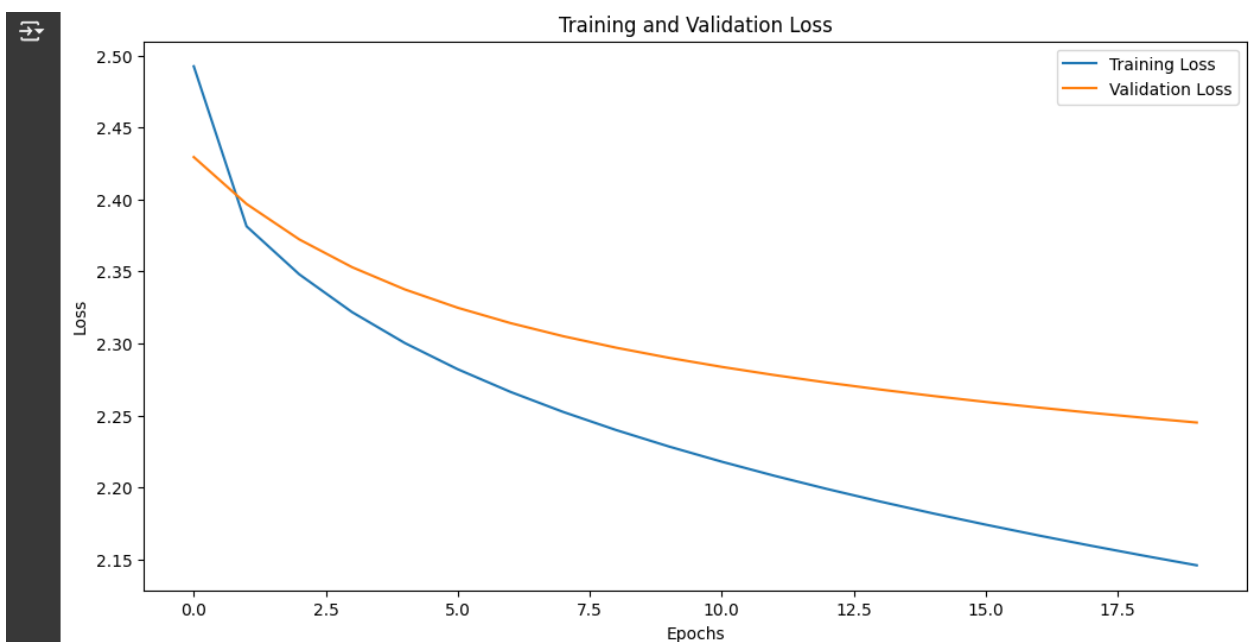


Рисунок 3.5 – Графік залежності втрат від кількості епох

У підсумку, реалізація базової моделі класифікації зображень без використання метаданих дозволила отримати початкові результати та встановити базовий рівень продуктивності. Це створило основу для подальших експериментів з додаванням метаданих та впровадженням методів покращення якості класифікації, що є основними цілями даного дослідження.

### 3.2.3 Додавання метаданих до моделі класифікації зображень

Наступним етапом дослідження було поступове включення різних метаданих у модель та оцінка їх впливу на точність, функцію втрат та час класифікації, щоб визначити, наскільки метадані можуть покращити результати класифікації.

У процесі підготовки даних було розширено функцію попередньої обробки, щоб вона, окрім зображень та міток класів, також повертала метадані, пов'язані з об'єктами на зображеннях. Використані метадані включали:

- `is_difficult` – бінарний індикатор, який вказує, чи є об'єкт складним для розпізнавання;
- `is_truncated` – бінарний індикатор, що свідчить про те, чи об'єкт частково прихований або обрізаний на зображенні;
- `pose` – категоріальна змінна, яка описує позу об'єкта;
- координати обмежувальної рамки – числові значення, що визначають положення об'єкта на зображенні;
- кількість об'єктів на зображенні – числова змінна, що відображає загальну кількість об'єктів;
- площа обмежувальної рамки – числова змінна, що представляє площу об'єкта на зображенні;

– мітки класів об'єктів – числові значення, що відповідають класам об'єктів.

Ці метадані були об'єднані у вектор фіксованої розмірності, який використовувався як додатковий вхід до моделі. Таке представлення метаданих дозволило моделі обробляти їх у числовій формі та використовувати у подальшому навчанні.

Для інтеграції метаданих у модель класифікації зображень була розроблена архітектура, яка складається з двох паралельних гілок: обробки зображень та обробки метаданих.

Гілка обробки зображень базується на попередньо навченій моделі ResNet50, з якої видалено верхні шари класифікації. Ця модель використовується як потужний екстрактор ознак, що перетворює вхідні зображення у високорівневі представлення. Шари базової моделі були заморожені, щоб зберегти попередньо навчені ваги та уникнути перенавчання на нових даних. Після базової моделі застосовано шар глобального усереднення ознак, який зменшує розмірність виходу та створює вектор ознак фіксованої довжини.

Гілка обробки метаданих починається з вхідного шару, який приймає вектор метаданих. Цей вектор проходить через повнозв'язний шар з 16 нейронами та функцією активації ReLU. Вибір 16 нейронів був здійснений з метою забезпечити достатню здатність моделі навчатися з метаданих без надмірного ускладнення та ризику перенавчання. Цей шар виконує нелінійне перетворення метаданих, що дозволяє моделі виявляти складні зв'язки та взаємозалежності між різними метаданими та класами об'єктів.

Після отримання ознак з обох гілок вони були об'єднані за допомогою шару конкатенації. Об'єднаний вектор містить як візуальні ознаки зображення, так і перетворені метадані, що дозволяє моделі враховувати додаткову інформацію при класифікації. До об'єданого вектору було застосовано фінальний класифікаційний шар, який складається з повнозв'язного шару з кількістю нейронів, що відповідає кількості класів у

наборі даних, та функцією активації softmax. Це забезпечує отримання ймовірностей належності до кожного класу.

Процес навчання та аналізу результатів був аналогічним до базової моделі. Модель була скомпільована з використанням оптимізатора Adam з коефіцієнтом навчання 0,001, функції втрат «sparse\_categorical\_crossentropy» та метрики точності. Навчання проводилося протягом 20 епох на тренувальному наборі даних з валідацією на валідаційному наборі після кожної епохи. Під час навчання було виміряно час, витрачений на тренування моделі, щоб оцінити вплив додавання метаданих на загальну продуктивність процесу.

Після завершення навчання модель була протестована на валідаційному наборі даних, де були обчислені функція втрат та точність. Аналіз результатів проводився за допомогою тих самих валідаційних зображень та графіків, що і для базової моделі. Це забезпечило коректне порівняння та дозволило виявити відмінності у продуктивності моделей.

### 3.2.4 Реалізація методів покращеної роботи з метаданими

У цьому розділі описано практичну реалізацію методів, спрямованих на покращення роботи з метаданими в задачі класифікації зображень. Окремо були впроваджені такі методи:

- adaptive weight combination of modalities (AWCM) – адаптивна вагова комбінація модальностей;
- cross-modality interaction (CMI) – взаємодія між модальностями;
- metadata attention mechanism (MAM) – механізм уваги до метаданих;
- semantic feature generation (SFG) – генерація семантичних ознак.

Впровадження кожного методу окремо (рис. 3.6) дозволило оцінити їхній індивідуальний вплив на процес класифікації зображень.

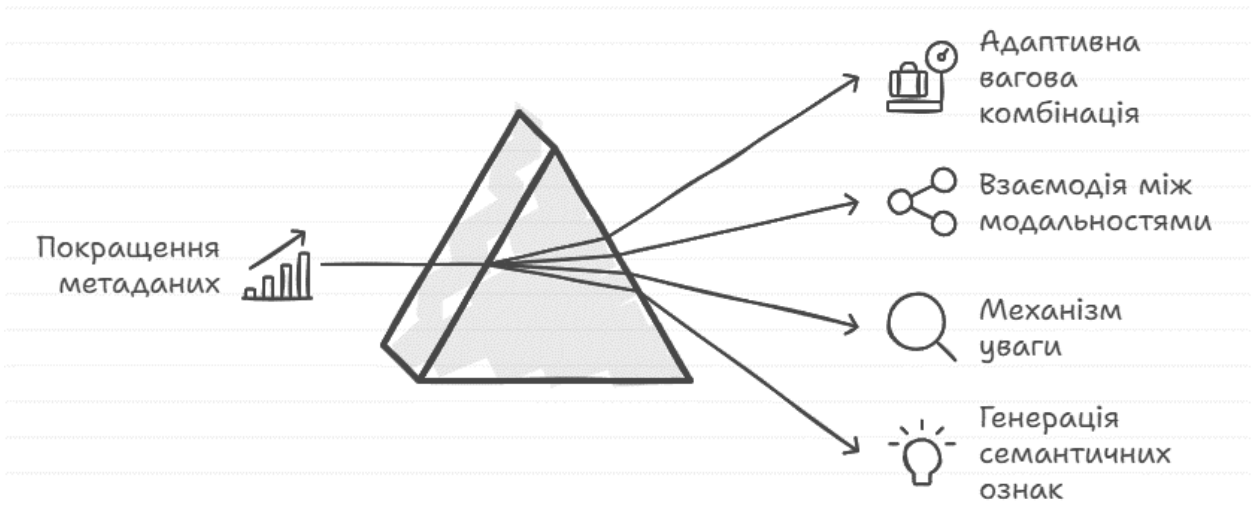


Рисунок 3.6 – Реалізовані методи покращеної роботи з метаданими

Для реалізації методу адаптивної вагової комбінації модальностей було модифіковано архітектуру нейронної мережі, додавши механізм адаптивного зважування між ознаками зображення та метаданих. Після обробки зображень за допомогою попередньо навченої моделі ResNet50 та отримання візуальних ознак, а також після перетворення метаданих через повнозв'язний шар, був впроваджений ваговий механізм. Далі ознаки зображення та метаданих множаться на відповідні ваги, що дозволяє масштабувати їхній внесок у фінальну класифікацію. Зважені ознаки об'єднуються та передаються на фінальний класифікаційний шар.

У реалізації методу взаємодії між модальностями до моделі було додано механізм, який дозволяє явно враховувати взаємодію між ознаками зображення та метаданих. Після отримання ознак з обох гілок виконувалося поелементне множення векторів ознак зображення та метаданих. Це дозволило захопити взаємозв'язки між модальностями на рівні ознак. Отриманий вектор взаємодії був об'єднаний з початковими ознаками зображення та метаданих шляхом конкатенації, утворюючи розширений вектор ознак. Цей комбінований вектор передавався на фінальний класифікаційний шар для прийняття рішення.

Для впровадження механізму уваги до метаданих до моделі було додано шар уваги, спеціально призначений для обробки метаданих. Після перетворення метаданих через повнозв'язний шар отриманий вектор ознак метаданих пропускався через додатковий повнозв'язний шар з функцією активації softmax. Це дозволило обчислити ваги уваги для кожного елемента вектора метаданих, які сумуються до 1. Отримані ваги уваги використовувалися для поелементного зважування ознак метаданих, що підсилювало вплив найбільш релевантних ознак та зменшувало вплив менш важливих. Зважений вектор метаданих об'єднувався з ознаками зображення, і отриманий вектор передавався на фінальний класифікаційний шар.

У реалізації методу генерації семантичних ознак обробка метаданих була розширена для створення додаткових семантично значущих ознак, які можуть бути корисними для класифікації. Це передбачало додавання додаткових шарів для обробки метаданих, таких як шар вбудовування або більш складні перетворення, щоб витягнути вищорівневу семантичну інформацію з метаданих. Згенеровані семантичні ознаки об'єднувалися з ознаками зображення, зазвичай шляхом конкатенації, і комбінований вектор використовувався у фінальному класифікаційному шарі.

Впровадження кожного методу вимагало модифікації архітектури моделі та процесу обробки даних. Усі зміни були здійснені в межах фреймворку TensorFlow з використанням високорівневого API Keras, що забезпечило зручність реалізації та експериментування. Кожен метод був реалізований окремо, зберігаючи основну структуру моделі, щоб забезпечити коректність порівняння результатів.

### 3.2.5 Реалізація комбінування методів роботи з метаданими

Наступним кроком було здійснено практичну реалізацію комбінування методів покращеної роботи з метаданими з метою дослідити, як їхнє спільне

використання впливає на класифікацію зображень – позитивно чи негативно. Зокрема, були скомбіновані такі методи, як адаптивна вагова комбінація модальностей, взаємодія між модальностями та генерація семантичних ознак. Реалізація цих методів у поєднанні дозволила дослідити їхній сумарний вплив на ефективність моделі та виявити можливі взаємозв'язки між ними.

На практиці це було досягнуто шляхом інтеграції компонентів кожного методу в архітектуру нейронної мережі. Комбінування цих методів дозволило провести експерименти та оцінити, як їхнє спільне застосування впливає на результати класифікації. Це дало змогу визначити, чи підсилюють методи один одного, чи, можливо, їхнє поєднання не дає очікуваного покращення або навіть негативно впливає на точність. Таким чином, практична реалізація комбінування методів стала важливим кроком у дослідженні їхнього комплексного впливу на ефективність класифікації зображень.

### 3.3 Інструкція користувача

У цьому розділі представлено інструкцію користувача для запуску та використання реалізованих моделей класифікації зображень з використанням метаданих та покращених методів їхньої обробки у середовищі Google Colab. Інструкція описує необхідні кроки для відкриття файлу, налаштування середовища, запуску моделей та інтерпретації результатів у межах Google Colab. Також розглянуто встановлення необхідних бібліотек та пакетів, а також налаштування доступу до GPU для підвищення швидкості обробки.

Основні кроки для запуску та використання реалізованих моделей:

Крок 1. Для початку роботи необхідно мати обліковий запис Google та доступ до Інтернету. Відкрийте браузер і перейдіть на Google Colab. Після входу в обліковий запис завантажте файл `.ipynb` на свій Google Drive. Для цього перетягніть файл у вікно браузера з відкритим Google Drive або

скористайтесь кнопкою «Нова» і виберіть «Завантажити файл», після чого виберіть відповідний файл зі свого комп'ютера (рис. 3.7).

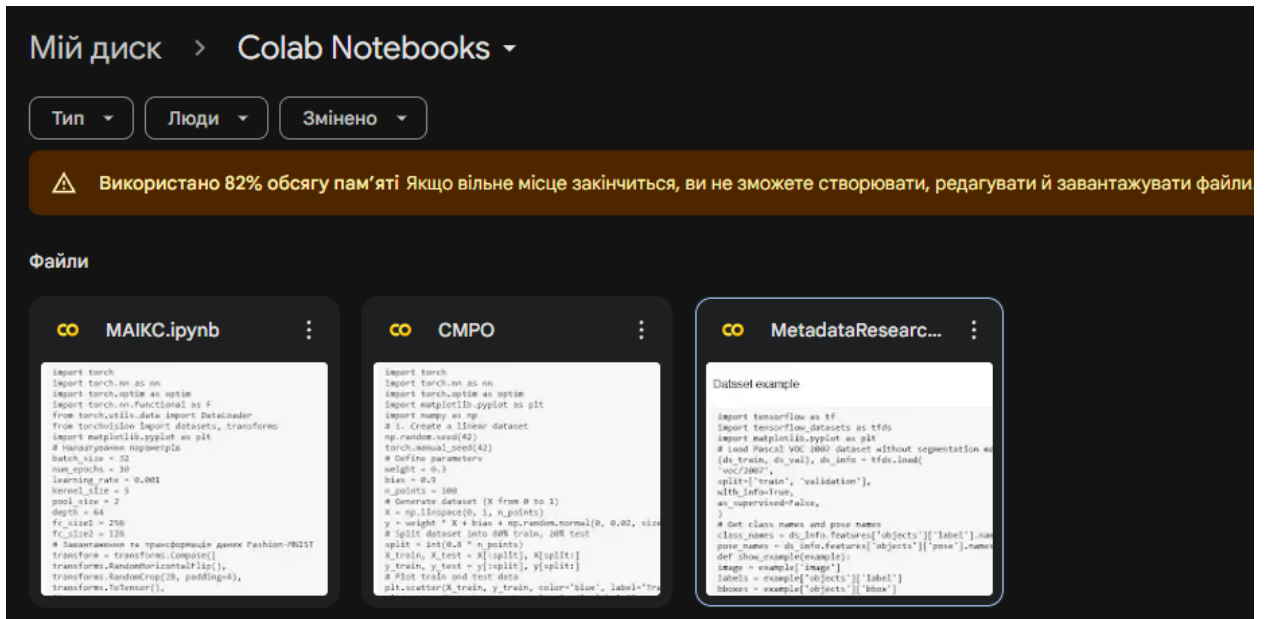


Рисунок 3.7 – Файли з проектами для Google Colab

Крок 2. Після завантаження файлу на Google Drive знайдіть його та відкрийте за допомогою Google Colab. Для цього клацніть правою кнопкою миші на файлі, оберіть пункт «Відкрити за допомогою» і виберіть «Google Colaboratory». Якщо такої опції немає, натисніть «Підключити додатки» і додайте Google Colab до списку доступних застосунків (рис. 3.8).

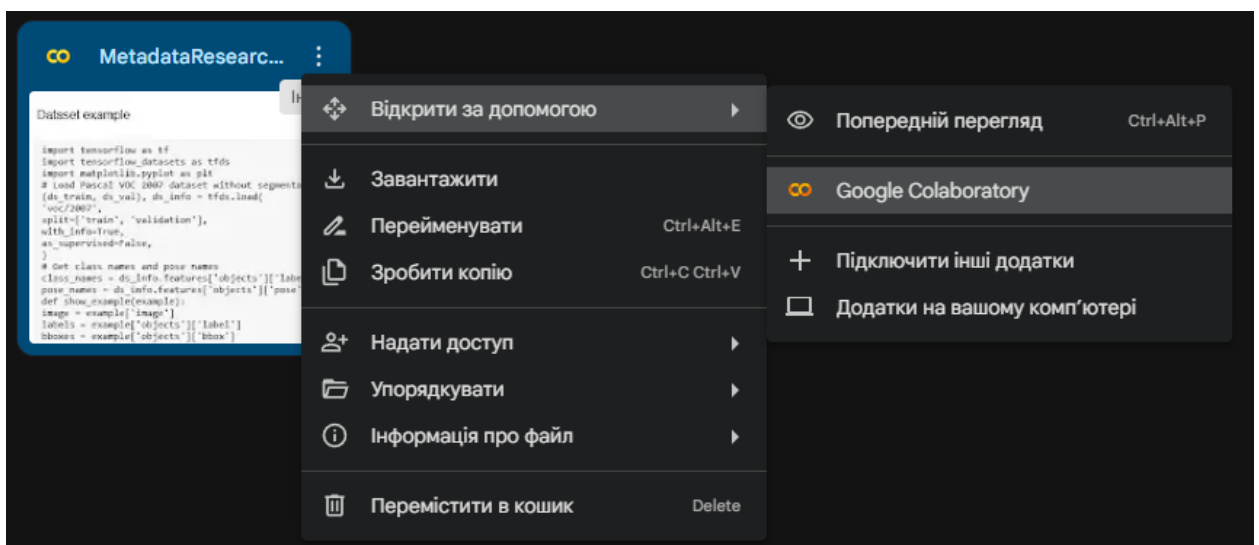


Рисунок 3.8 – Відкриття файлу у Google Colab

Крок 3. Відкривши файл у Google Colab, необхідно налаштувати середовище виконання. У верхньому правому куті натисніть кнопку «Підключитися» для встановлення з'єднання з сервером. Далі оберіть «Середовище виконання» у головному меню та натисніть «Змінити тип середовища виконання». У вікні, що з'явиться, виберіть апаратний прискорювач «GPU» з випадаючого списку (рис. 3.9). Це дозволить використовувати графічний процесор для прискорення навчання моделей. Після вибору натисніть «Зберегти».

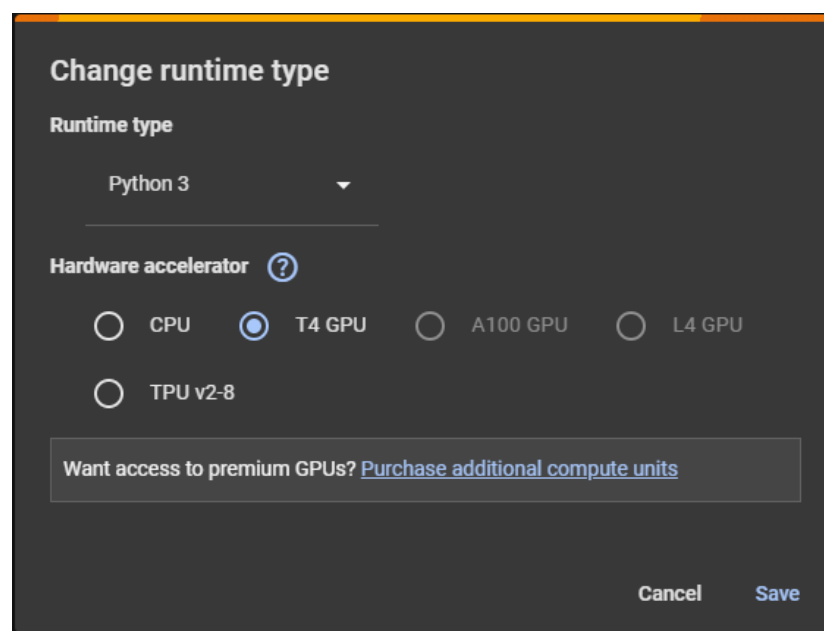


Рисунок 3.9 – Налаштування середовища виконання

Крок 4. Проект складається з комірок з кодом та текстом, які структуровані за розділами. Комірки з кодом містять реалізацію моделей та необхідні функції, а комірки з текстом надають пояснення та інструкції. Перед початком роботи рекомендується ознайомитися зі структурою проекту (рис. 3.10), щоб зрозуміти послідовність виконання коду та логіку побудови моделей.

Крок 5. Для виконання коду необхідно послідовно запускати комірки зверху вниз. Для цього клацніть на комірці та натисніть клавішу Shift + Enter або натисніть кнопку старту ліворуч від комірки. У деяких випадках може

знадобитися встановити додаткові бібліотеки. Це можна зробити, виконавши відповідні команди в комірках з кодом, наприклад, `!pip install tensorflow`.

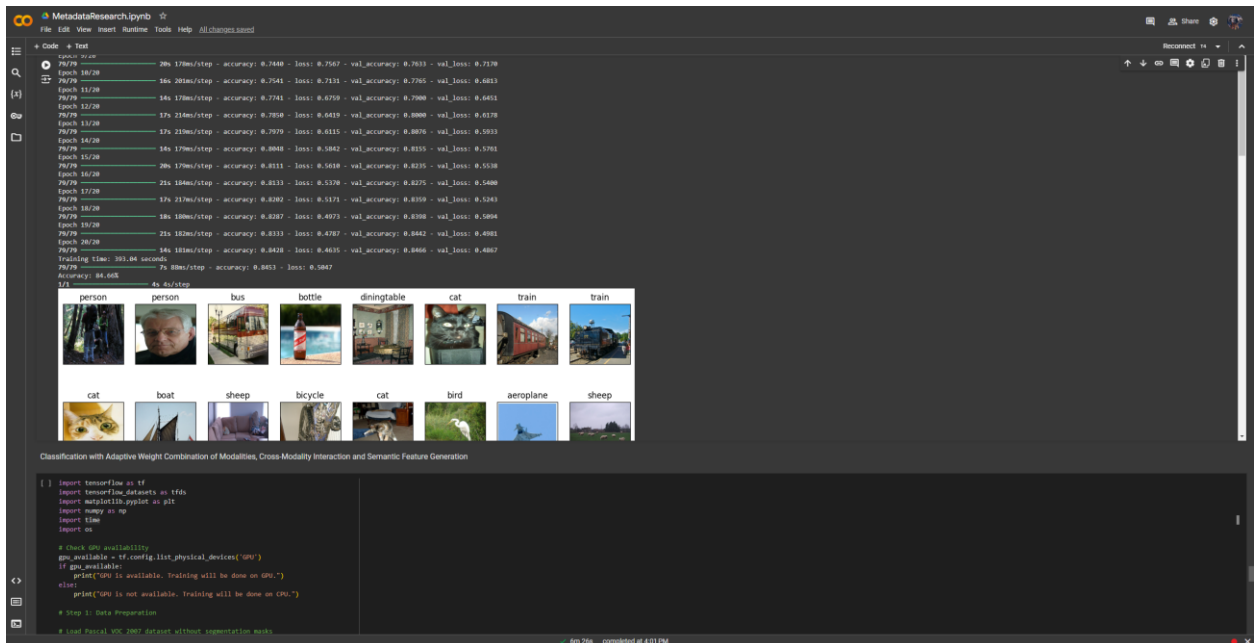


Рисунок 3.10 – Структура проєкту

Пам'ятайте про обмеження Google Colab щодо часу виконання та використання ресурсів. Рекомендується періодично зберігати ноутбук та результати роботи на Google Drive або завантажувати їх на локальний комп'ютер, щоб уникнути втрати даних. Якщо виникають проблеми з підключенням до GPU або іншими ресурсами, можна спробувати переключитися на інший апаратний прискорювач або тимчасово працювати без нього.

### 3.4 Тестування розробленої моделі

Тестування проводилося на спеціально підготовленому наборі даних, де оцінювалася продуктивність моделей за такими метриками: фінальна валідаційна точність, найкраща валідаційна точність, епоха досягнення найкращої точності та час тренування (табл. 3.1).

Результати тестування свідчать про суттєвий вплив використання метаданих та покращених методів їх обробки на точність класифікації зображень. Базова модель без метаданих показала фінальну валідаційну точність 34,46% (рис. А.2), що є найнижчим показником серед усіх протестованих моделей. Додавання метаданих поступово покращує точність моделі. Так, модель з 2 параметрами метаданих досягає 37,73% (рис. А.5), з 4 параметрами – 39,88% (рис. А.8), з 6 параметрами – 41,55% (рис. А.11), а з 7 параметрами метаданих – вже 71,59% (рис. А.14).

Таблиця 3.1 – Результати тестування

<b>Модель</b>	<b>Фінальна валідаційна точність</b>	<b>Найкраща валідаційна точність</b>	<b>Епоха найкращої точності</b>	<b>Час тренування (секунди)</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
Без метаданих	34,46%	34,58%	17	413,55
3 метаданими (2 параметри)	37,73%	37,73%	20	433,59
3 метаданими (4 параметри)	39,88%	39,88%	20	491,23
3 метаданими (6 параметрів)	41,55%	41,55%	20	481,40
3 метаданими (7 параметрів)	71,59%	71,59%	20	390,04
3 використанням AWCM	84,82%	84,82%	20	390,93
3 використанням СМІ	83,55%	83,55%	20	383,16

Продовження таблиці 3.1

1	2	3	4	5
3 використанням MAM	74,30%	74,30%	20	399,11
3 використанням SFG	87,97%	88,01%	19	377,29
3 поєднанням AWCM та CMI	84,82%	84,82%	20	383,35
3 поєднанням AWCM, CMI та SFG	84,86%	84,86%	20	402,31

Особливо показовим є те, що модель з 7 параметрами метаданих досягає значно вищої точності у порівнянні з моделями, які використовують меншу кількість метаданих. Це різке покращення пояснюється тим, що сьомий параметр метаданих є маркером класу об'єкта. Тобто, цей параметр безпосередньо вказує на категорію, до якої належить зображення, що суттєво полегшує завдання класифікації для моделі. Цей результат свідчить про те, що на продуктивність моделі впливає не лише кількість використаних метаданих, але й їхній тип та інформативність. Метадані, які містять релевантну та значущу інформацію про об'єкт, можуть значно покращити точність класифікації. Наприклад, маркер класу фактично надає моделі підказку щодо правильного результату, що пояснює значне підвищення точності. Таким чином, важливо не тільки збільшувати кількість метаданих, але й забезпечувати, щоб вони були максимально корисними та релевантними для конкретної задачі. Вибір інформативних метаданих, які безпосередньо пов'язані з цільовим класом або ознаками об'єкта, може мати вирішальний вплив на ефективність моделі. Це підкреслює необхідність

ретельного відбору та аналізу метаданих при розробці моделей машинного навчання.

Покращені методи обробки метаданих окремо показали значне підвищення точності моделі. Реалізація адаптивної вагової комбінації модальностей (AWCM) дозволила досягти фінальної валідаційної точності 84,82% (рис. А.17). Це свідчить про те, що адаптивне поєднання візуальних ознак та метаданих, при якому модель визначає вагу кожної модальності, покращує здатність моделі вловлювати важливу інформацію з обох джерел.

Взаємодія між модальностями (СМІ) також окремо підвищила точність моделі до 83,55% (рис. А.20). Цей метод дозволяє моделі встановлювати зв'язки між ознаками зображення та метаданими, що сприяє кращому розумінню контексту та більш точній класифікації.

Механізм уваги до метаданих (МММ) досягнув точності 74,30% (рис. А.23). Використання механізму уваги дозволяє моделі фокусуватися на найбільш релевантних метаданих для кожного зображення, що покращує якість класифікації шляхом зменшення впливу несуттєвої інформації. Варто зазначити, що механізм уваги до метаданих досягнув незначного приросту порівняно з моделлю без покращень. Це свідчить про те, що МММ не повністю реалізував свій потенціал у контексті обраного набору даних та підібраних метаданих.

Найвищої точності 87,97% (рис. А.26) було досягнуто при використанні генерації семантичних ознак (SFG). Цей метод дозволяє моделі генерувати більш інформативні ознаки на основі метаданих, що покращує розуміння моделі про об'єкти на зображеннях та їхні характеристики.

Однак, поєднання цих методів між собою не призвело до значного приросту точності, а інколи не мало впливу взагалі. Наприклад, модель, яка поєднує адаптивну вагову комбінацію модальностей та взаємодію між модальностями (AWCM та СМІ), досягає тієї ж точності 84,82% (рис. А.29), що й модель лише з AWCM. Це може бути пов'язано з тим, що обидва методи намагаються оптимізувати поєднання візуальних ознак та метаданих,

але роблять це різними способами. Їх одночасне використання призвело до перенасичення моделі подібною інформацією або до конфлікту між механізмами, що відповідають за інтеграцію модальностей.

У фінальному поєднанні, де були об'єднані три покращення (AWCM, CMI та SFG), модель показала точність 84,86% (рис. А.32), що є нижчим результатом порівняно з використанням лише генерації семантичних ознак. Це вказує на те, що додавання інших методів до SFG не лише не покращило, але й погіршило продуктивність моделі. На мою думку, це пов'язано з тим, що SFG вже ефективно захоплює необхідну інформацію з метаданих, і додаткові методи створюють надлишковість або навіть шум, що ускладнює навчання моделі. Крім того, складніші моделі можуть бути більш схильними до перенавчання, особливо якщо обсяг даних обмежений.

Таким чином, хоча окремі покращені методи обробки метаданих демонструють суттєве підвищення точності, їх комбінування не завжди призводить до додаткових переваг. Це підкреслює важливість ретельного підбору та тестування методів обробки метаданих при розробці моделей глибокого навчання. Можливо, варто зосередитися на оптимізації окремих методів або розробці нових підходів, які більш ефективно інтегрують різні модальності без негативного впливу на продуктивність моделі.

Для більш детального аналізу результатів були побудовані різноманітні графіки.

Графік порівняння валідаційної точності між моделями (рис. 3.11) відображає зміну валідаційної точності кожної моделі протягом епох тренування. На осі X відкладені епохи, а на осі Y – валідаційна точність у відсотках. Кожна лінія відповідає певній моделі. Графік показує, що моделі з використанням метаданих та покращених методів обробки демонструють більш швидке та стабільне зростання точності протягом тренування. Особливо це помітно для моделей з SFG, AWCM та CMI.

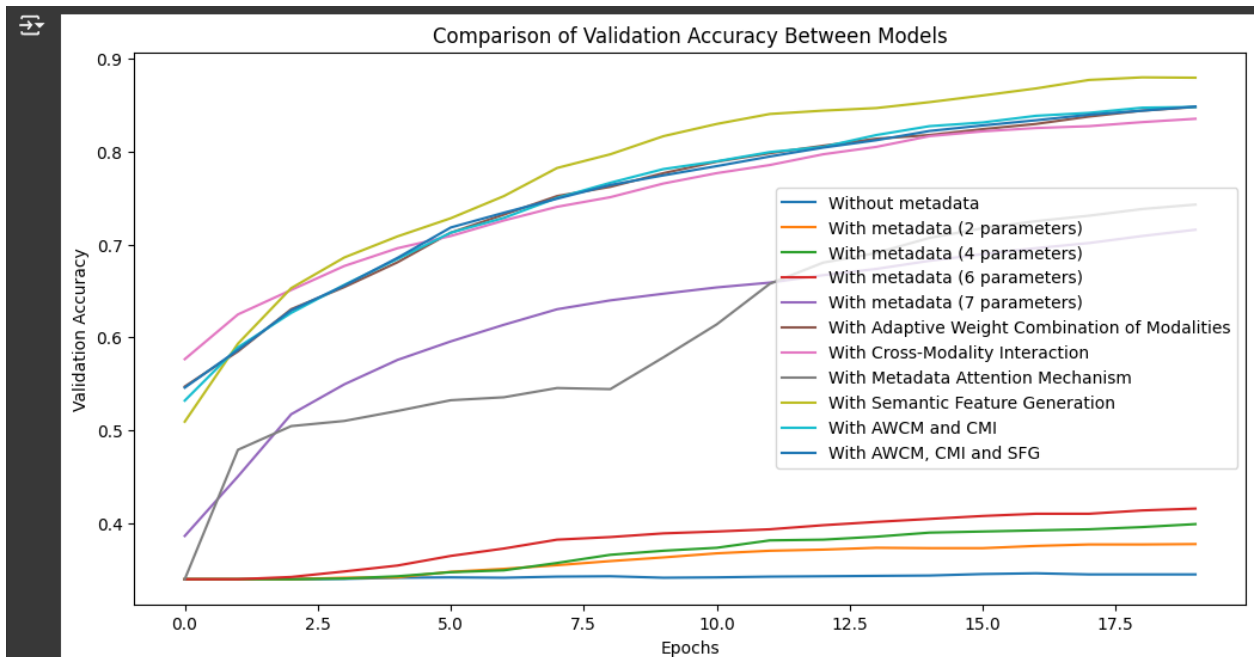


Рисунок 3.11 – Графік порівняння валідаційної точності

Графік порівняння валідаційних втрат між моделями (рис. 3.12) ілюструє зміну валідаційних втрат для кожної моделі за епохами. Зниження втрат свідчить про покращення моделі. З графіка видно, що моделі з покращеними методами обробки метаданих мають швидше зниження валідаційних втрат, що вказує на ефективніше навчання та краще узгодження з даними.

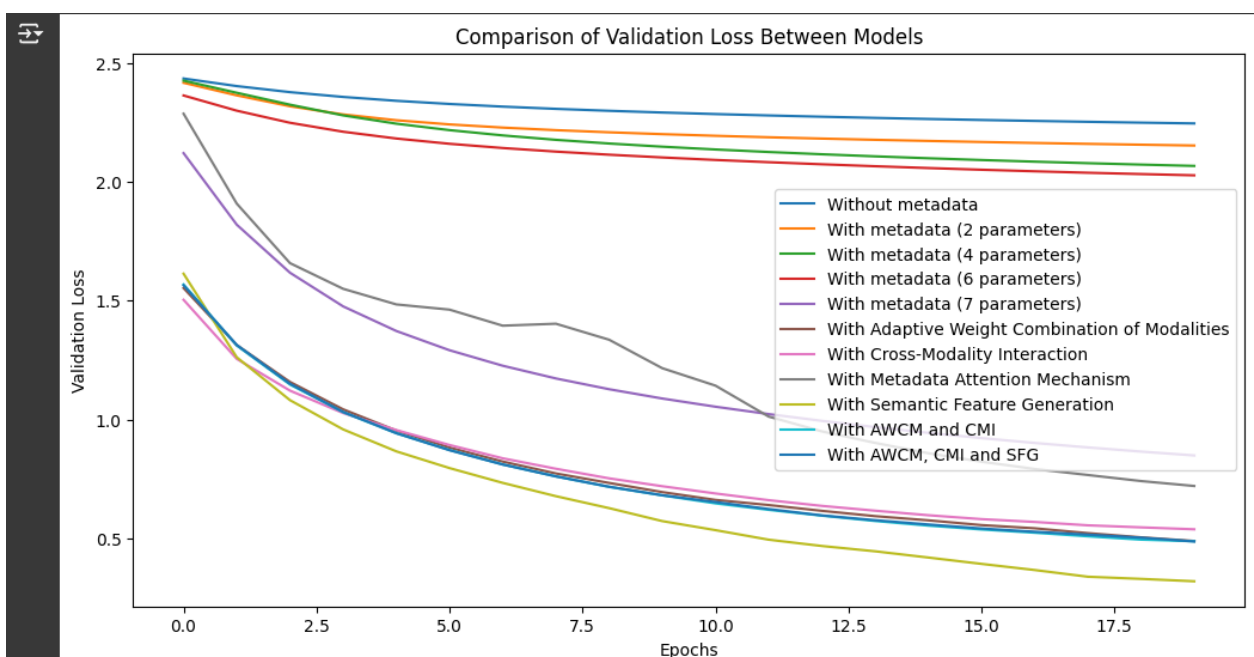


Рисунок 3.12 – Графік порівняння валідаційних втрат

Стовпчиковий графік порівняння часу тренування між моделями (рис. 3.13) демонструє час тренування кожної моделі в секундах. Це дозволяє оцінити обчислювальну ефективність моделей. З графіка видно, що моделі з покращеними методами не завжди вимагають більше часу на тренування, оскільки ці методи можуть підвищувати ефективність навчання. Наприклад, метод генерації семантичних ознак, надає моделі більш релевантні та інформативні ознаки, що сприяє швидшій збіжності та зменшенню кількості необхідних обчислень. Це дозволяє моделі досягти високої точності за меншу кількість епох або з меншими витратами часу на кожну епоху. Отже, впровадження покращених методів може зробити модель більш ефективною як з точки зору точності, так і обчислювальних витрат, що пояснює, чому вони не завжди потребують більше часу на тренування.

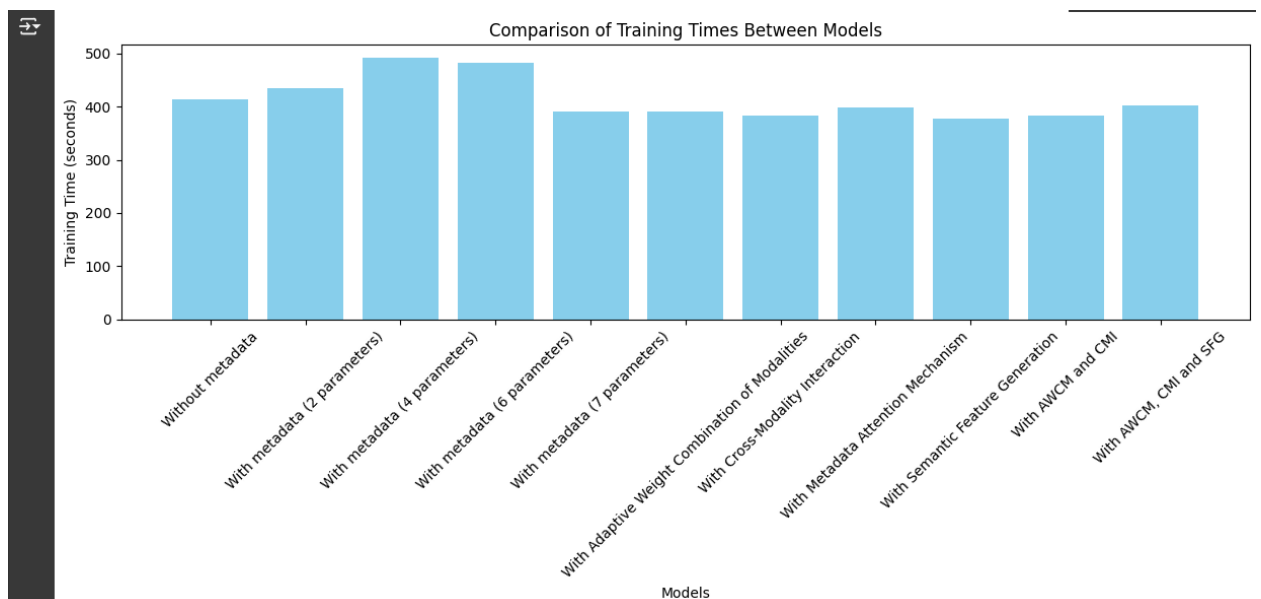


Рисунок 3.13 – Графік порівняння часу тренування

Стовпчиковий графік на рисунку 3.14 порівнює фінальну валідаційну точність усіх моделей. Він наочно демонструє, яка модель досягла найвищої точності після завершення тренування. Графік показує, що модель з SFG має найвищу фінальну валідаційну точність. Інші моделі з покращеними методами обробки метаданих також демонструють високі результати, значно перевершуючи базову модель без метаданих.

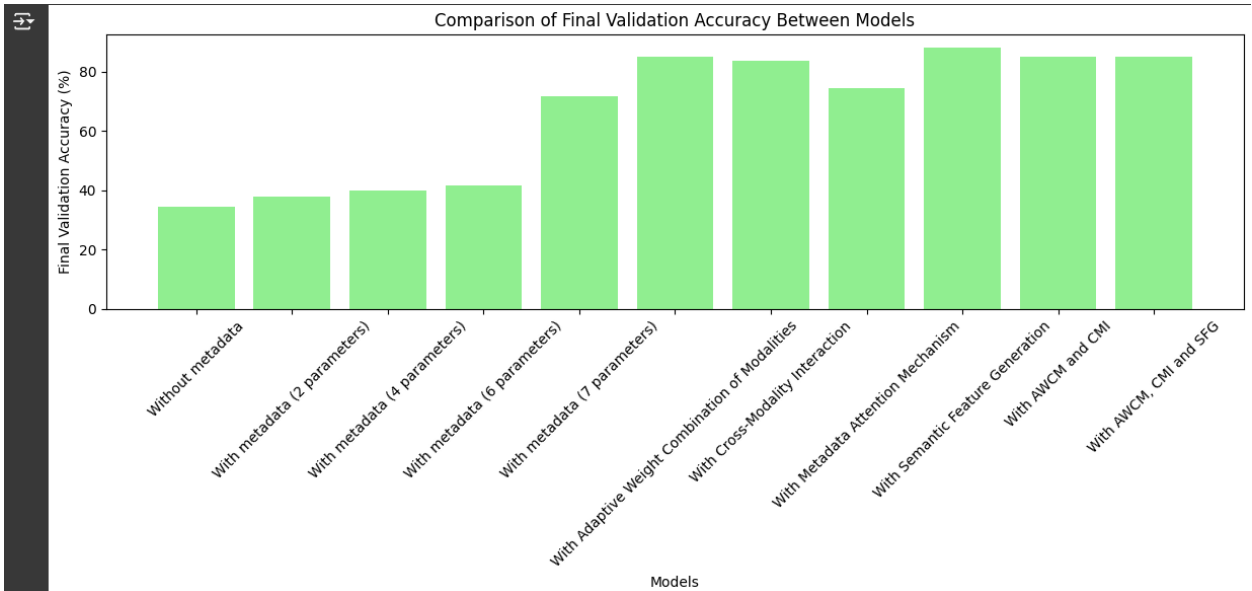


Рисунок 3.14 – Графік порівняння фінальної валідаційної точності

Графік на рисунку 3.15 відображає найкращу валідаційну точність, досягнуту кожною моделлю протягом тренування. Це дозволяє оцінити потенціал моделей та їх максимальну продуктивність. Графік підтверджує, що модель з SFG досягає найвищої найкращої точності – 88,01%, що майже відповідає її фінальній точності, свідчаючи про стабільність моделі.

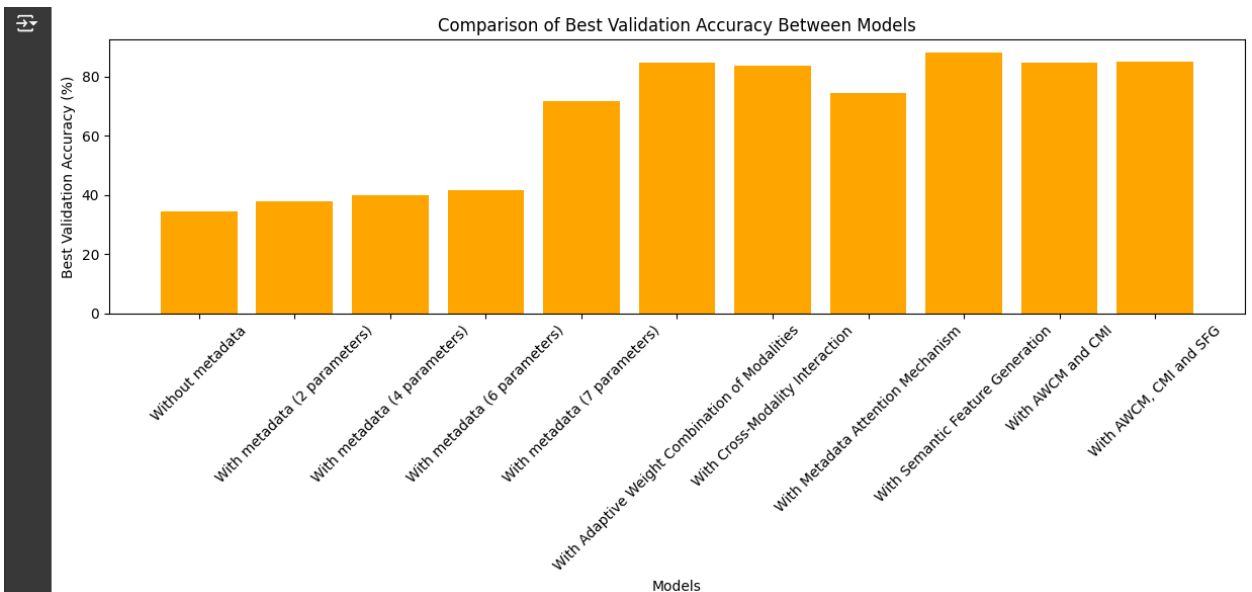


Рисунок 3.15 – Графік порівняння найкращої валідаційної точності

На рисунку 3.16 показано графік залежності фінальної валідаційної точності моделей від часу їх тренування. Це допомагає оцінити ефективність

моделей з точки зору витрат часу та досягнутої точності. З графіка видно, що моделі з високою точністю не обов'язково вимагають більше часу на тренування. Наприклад, модель з SFG досягає найвищої точності за відносно короткий час, що свідчить про її обчислювальну ефективність.

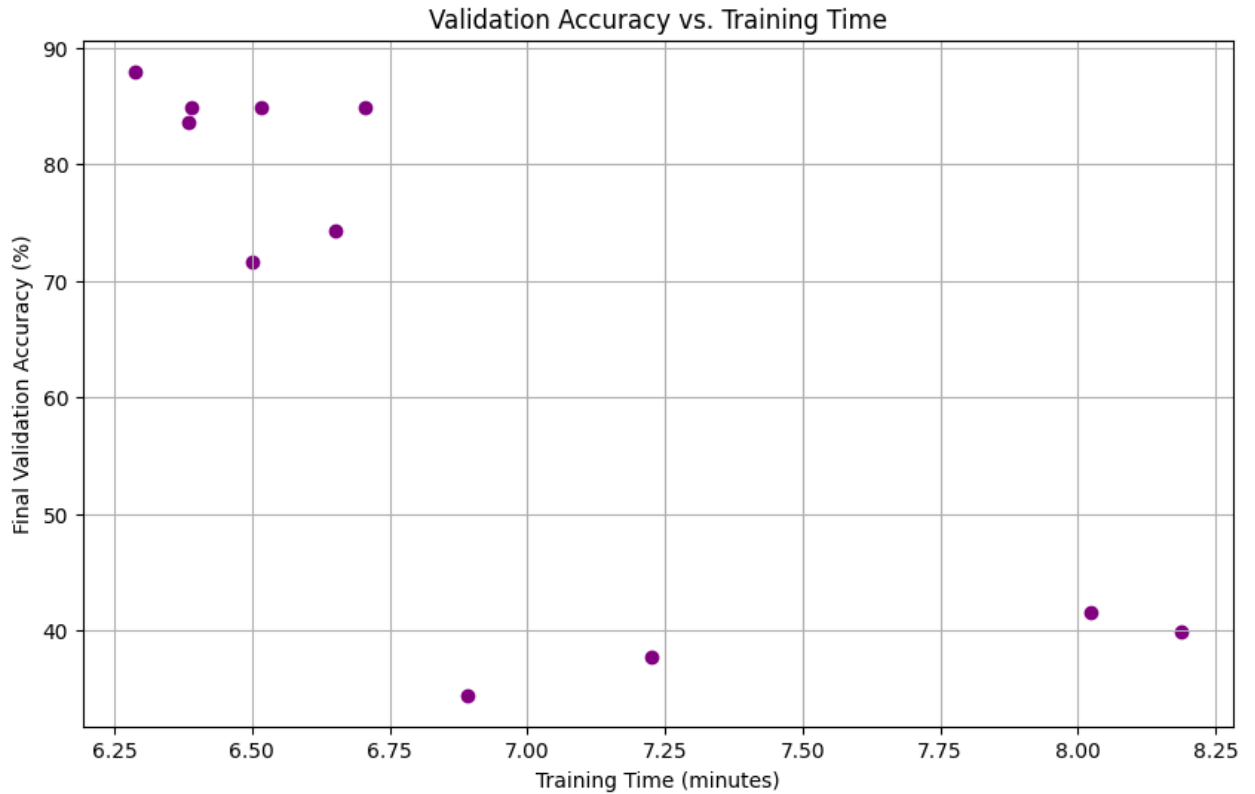


Рисунок 3.16 – Графік залежності валідаційної точності від часу тренування

## ВИСНОВКИ

У рамках кваліфікаційної роботи було досліджено інтеграцію метаданих у процес класифікації зображень з використанням глибоких нейронних мереж. Було розглянуто та впроваджено різні підходи до інтеграції метаданих, включаючи адаптивну вагову комбінацію модальностей, взаємодію між модальностями, механізм уваги до метаданих та генерацію семантичних ознак. Крім того, були розглянуті комбіновані методи інтеграції, що дозволяють максимально використовувати переваги кожного з них.

У результаті дослідження здійснено програмну реалізацію моделей інтеграції метаданих, яка може бути використана для подальших досліджень та практичних застосувань у різних галузях комп'ютерного бачення.

Експериментальне тестування показало, що інтеграція метаданих суттєво підвищує точність класифікації зображень порівняно з базовою моделлю без метаданих. Метод генерації семантичних ознак показав найвищу ефективність, забезпечуючи створення більш глибоких та інформативних ознак, що суттєво покращує здатність моделі до розпізнавання складних патернів. Також було проаналізовано обчислювальну ефективність розроблених методів, що дозволило визначити баланс між точністю класифікації та вимогами до ресурсів моделі.

Комбінування методів інтеграції метаданих не призвело до очікуваного підвищення точності класифікації. У деяких випадках спостерігалось навіть зниження продуктивності порівняно з використанням окремих методів, що може бути пов'язано з надмірною складністю моделі або конфліктом між різними підходами до інтеграції інформації. Це підкреслює важливість ретельного підбору та налаштування методів інтеграції для досягнення оптимальних результатів.

Проте, попри досягнуті результати, існують певні обмеження, такі як можливість перенавчання при надмірній кількості метаданих та необхідність

ретельного налаштування параметрів моделей при комбінуванні різних методів. В майбутньому планується розширити дослідження, включивши більш складні методи інтеграції метаданих та протестувати їх на різних наборах даних для оцінки їх універсальності та адаптивності.

Наукова новизна дослідження полягає в огляді комбінування методів, що базуються на генерації семантичних ознак, а також методів, що орієнтовані на роботу з модальностями.

Таким чином, проведені дослідження підтвердили важливість інтеграції метаданих у процес класифікації зображень, що дозволяє створювати більш точні та ефективні моделі, здатні успішно застосовуватись у різних практичних задачах комп'ютерного бачення. Розроблені методи інтеграції метаданих відкривають нові можливості для подальшого вдосконалення систем розпізнавання та класифікації зображень, що має велике значення для розвитку сучасних технологій штучного інтелекту.

Результати дослідження апробовано у вигляді тез доповідей під час X Міжнародної науково-практичної конференції «Computer-integrated technologies of automation of technological processes» [46].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.
2. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.
3. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.
4. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.
5. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.
6. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. *International Journal of Advanced Trends in Computer Science and Engineering*. 9(4). – 4701-4706.
7. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

8. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET) (pp. 665-670). IEEE.
9. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.
10. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069-6084.
11. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.
12. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.
13. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform.
14. Lyashenko, V., Mohammad, A., & Kobylin, O. (2015). Experiments with Fusion of Images with Use of Wavelet Transformation in Problems of the Text Information Analysis.
15. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. Системи управління, навігації та зв'язку. Збірник наукових праць, 5(57).
16. Путятін, Є. П., Гороховатський, В. О., & Матат, О. О. (2006). Методи та алгоритми комп'ютерного зору: навч. посібник.
17. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.

18. Daradkeh, Y.I., Gorokhovatskiy, V., Tvoroshenko, I., Gadetska, S., and Al-Dhaifallah, M. (2021) Methods of Classification of Images on the Basis of the Values of Statistical Distributions for the Composition of Structural Description Components, *IEEE Access*, 9, pp. 92964-92973.

19. Gorokhovatskiy, V.O., Tvoroshenko, I.S., and Peredrii O.O. (2020) Image classification method modification based on model of logic processing of bit description weights vector, *Telecommunications and Radio Engineering*, 79(1), pp. 59-69.

20. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET) (pp. 930-935). IEEE.

21. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. *Information Technology and Management Science*, 19(1), 23-28.

22. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.

23. Kobylin, O., & Lyashenko, V. (2014). Comparison of standard image edge detection techniques and of method based on wavelet transform.

24. Gorokhovatskiy, V. A., Kobylin, O. A., & Kulikov, Y. A. (2015). Application of Granulation of Feature Descriptions in Structural Image Recognition. *Telecommunications and Radio Engineering*, 74(6).

25. Kuzminska, O., Mazorchuk, M., Morze, N., & Kobylin, O. (2019, June). Digital learning environment of ukrainian universities: The main components to influence the competence of students and teachers. In International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (pp. 210-230). Springer, Cham.

26. Kinoshenko, D., Kobylin, O., Mashtalir, S., & Stolbovyi, M. (2019, March). Metric video retrieval speedup by irrelevant data elimination. In Eleventh International Conference on Machine Vision (ICMV 2018) (Vol. 11041, pp. 176-183). SPIE.
27. Lyashenko, V., Matarneh, R., Kobylin, O., & Putyatin, Y. (2016). Contour detection and allocation for cytological images using Wavelet analysis methodology.
28. Yakovleva, O., Kovtunencko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System (COLINS-2023). In CEUR Workshop Proceedings (Vol. 3403, pp. 69-86).
29. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.
30. Творошенко, І. С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. *Харків: ХНУРЕ*.
31. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.
32. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.
33. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.
34. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.

35. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.
36. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.
37. Stiefel, M., Müller, M., Bachmann, B. I., Guitar, M. A., Nayak, U. P., & Mücklich, F. (2024). Enhancing machine learning classification of microstructures: A workflow study on joining image data and metadata in CNN. *MRS Communications*, 1-9.
38. Cai, Z., Song, J., Zhang, T., Hu, C., & Jing, X. Y. (2024). Local weight coupled network: multi-modal unequal semi-supervised domain adaptation. *Multimedia Tools and Applications*, 83(2), 4331-4357.
39. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. In *Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60- 63). ACM.
40. Huang, L., Peng, Z., Chen, F., Dai, S., He, Z., & Liu, K. (2024). Cross-modality interaction for few-shot multispectral object detection with semantic knowledge. *Neural Networks*, 173, 106156.
41. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2018, August). Representative Based Clustering of Long Multivariate Sequences with Different Lengths. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 545-548). IEEE.
42. Wang, P., Xie, Y., Li, X., & Li, Q. (2024). Allocation of attention to metadata and retrieval functions: Implications for perceived value and open data discovery and reuse. *Journal of Librarianship and Information Science*, 56(3), 607-622.
43. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). *Advances in Spatio-Temporal Segmentation of Visual Data* (Vol. 876). Springer Nature.

44. Wang, L., Dong, S., Chen, Y., Meng, X., Fang, S., & Fei, S. (2024). MetaSegNet: Metadata-collaborative Vision-Language Representation Learning for Semantic Segmentation of Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*.

45. Кобилін, О. А., & Путятіна, О. Є. (2024). Real-time image denoising corrupted by shot-noise. *Системи обробки інформації*, (1 (176)), 46-51.

46. Makhanov, R., Kobylin, O., (2024). Computer-integrated technologies of automation of technological processes with Proceedings of the X International Scientific and Practical Conference, November 05-08, 2024. Hamburg, Germany, pp. 67-74.