

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Оптимізація процесу класифікації та пошуку  
зображення цифрових об'єктів за допомогою методів  
машинного навчання»

на здобуття освітнього ступеня магістра  
зі спеціальності 121 Інженерія програмного забезпечення  
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання  
ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ Дмитро АДАМСЬКИЙ  
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-61  
Дмитро АДАМСЬКИЙ

Керівник: Вікторія КОРЕЦЬКА  
\_\_\_\_\_  
*канд. пед. наук, доц..*

Рецензент: \_\_\_\_\_  
*науковий ступінь,  
вчене звання* Ім'я, ПРІЗВИЩЕ

**Київ 2026**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ Ірина ЗАМРІЙ

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Адамському Дмитру Олексійовичу

1. Тема кваліфікаційної роботи: «Оптимізація процесу класифікації та пошуку зображення цифрових об'єктів за допомогою методів машинного навчання»

керівник кваліфікаційної роботи Вікторія КОРЕЦЬКА, канд. пед. наук, доц.,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 р. № 467.

2. Строк подання кваліфікаційної роботи «19» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, набір цифрових зображень; параметри нейронної мережі та алгоритмів оптимізації; метод вилучення та порівняння ознак; вимоги до точності класифікації та швидкодії системи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження методів класифікації та пошуку зображень у комп'ютерному зорі.

2. Аналіз сучасних архітектур глибокого навчання для вилучення та оптимізації простору ознак.
3. Розробка та валідація оптимізованої моделі класифікації і пошуку зображень на основі глибокого навчання.
5. Перелік ілюстративного матеріалу: *презентація*
  1. Моделі процесу оптимізації класифікації та пошуку зображень.
  2. Архітектура оптимізованого конвеєра обробки цифрових об'єктів.
  3. Послідовність етапів оптимізації базової моделі класифікації.
  4. Алгоритм ітеративної оптимізації з використанням EfficientNet-V0 та метричного навчання.
  5. Ілюстрація процесу pruning та побудови оптимізованої моделі.
  6. Порівняльна ефективність базової та оптимізованої моделей.
  7. Демонстрація практичного результату роботи системи класифікації та пошуку.
6. Дата видачі завдання «31» жовтня 2025 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	31.10-10.11.2025	
2	Вивчення матеріалів для аналізу методів класифікації та пошуку зображень.	11.11-14.11.2025	
3	Дослідження методів глибокого навчання для вилучення та оптимізації ознак.	15.11-18.11.2025	
4	Аналіз особливостей впливу методів оптимізації на якість класифікації та пошуку зображень.	19.11-25.11.2025	
5	Дослідження технологій формування ембедінгів та побудови простору ознак.	26.11.-30.11.2025	
6	Застосування методів глибокого навчання для оптимізації процесу класифікації та пошуку цифрових об'єктів.	1.12.-10.12.2025	
7	Оформлення роботи: вступ, висновки, реферат	11.12-16.12.2025	
8	Розробка демонстраційних матеріалів	17.12-19.12.2025	

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Дмитро АДАМСЬКИЙ

Керівник кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Вікторія КОРЕЦЬКА





## РЕФЕРАТ

Текстова частина магістерської роботи на здобуття освітнього ступеня магістра 94 стор., 1 табл., 20 рис., 30 джерел.

*Мета роботи* - Оптимізація процесу класифікації та пошуку зображень цифрових об'єктів шляхом удосконалення простору ознак та застосування методів машинного навчання, що забезпечують підвищення точності й зменшення обчислювальних витрат

*Об'єкт дослідження* - технологічний процес автоматизованої розподілу та семантичного пошуку серед масивів цифрових зображень з використанням алгоритмів машинного навчання.

*Предмет дослідження* - архітектурні рішення та алгоритмічні методи підвищення продуктивності систем комп'ютерного зору, спрямовані на вдосконалення якості та швидкості аналізу візуальних даних.

У роботі проведено аналіз сучасних проблем класифікації та пошуку цифрових об'єктів у системах комп'ютерного зору, а також обґрунтовано необхідність застосування оптимізаційних алгоритмів для підвищення ефективності обробки великомасштабних наборів зображень. Досліджено сучасні підходи до побудови нейронних мереж для розпізнавання та порівняння зображень, визначено їх переваги та недоліки в умовах зростання обсягів даних та високовимірності ознак.

Проаналізовано архітектури згорткових нейронних мереж, методи трансферного та метричного навчання, алгоритми формування компактних ембедінгів та методи підвищення структурованості простору ознак. Розглянуто техніки оптимізації моделей, зокрема pruning, quantization, fine-tuning та прискорені алгоритми пошуку найближчих сусідів, що дозволяють підвищити швидкодію системи без втрати точності.

Розроблено оптимізований конвеєр класифікації та пошуку зображень, який поєднує попередньо навчену модель EfficientNet-B0, формування векторних

представлень, нормалізацію та багатокрокову оптимізацію параметрів. Реалізовано методи метричного навчання для підвищення внутрішньокласової компактності та міжкласової роздільності, що суттєво підвищило точність пошуку за зображенням.

У рамках практичної частини створено експериментальну систему, яка виконує повний цикл обробки: класифікацію, генерацію ембедінгів, оптимізацію простору ознак, індексацію великої бази даних та пошук релевантних зображень у режимі offline. Проведено тестування та порівняльний аналіз базової та оптимізованої моделей, що підтвердило підвищення точності класифікації та значне скорочення часу пошуку.

Запропоновано рекомендації щодо подальшого вдосконалення систем класифікації та пошуку візуальних даних, зокрема перспективи застосування трансформерних архітектур, адаптивних оптимізаторів та методів автоматичного підбору гіперпараметрів. Визначено напрями подальших досліджень, пов'язаних зі збільшенням масштабованості системи, інтеграцією машинного навчання з фреймворками індексації та використанням апаратних прискорювачів.

Результати дослідження можуть бути використані для вдосконалення систем комп'ютерного зору, створення високопродуктивних сервісів пошуку за зображенням, розробки рекомендаційних систем, автоматизованих інструментів контролю якості та інтелектуальних систем аналізу великих масивів візуальних даних.

**КЛЮЧОВІ СЛОВА:** ОПТИМІЗАЦІЯ КЛАСИФІКАЦІЇ, МАШИННЕ НАВЧАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, КОМП'ЮТЕРНИЙ ЗІР, НЕЙРОННІ МЕРЕЖІ, ПОШУК ЗОБРАЖЕНЬ

## ABSTRACT

Text part of the master's qualification work: 94 pages, 20 pictures, 1 table, 30 sources.

The purpose of the work - optimization of the process of image classification and retrieval of digital objects by improving the feature space and applying machine learning methods that increase accuracy while reducing computational costs.

Object of research - the technological process of automated distribution and semantic retrieval within large collections of digital images using machine learning algorithms.

Subject of research - architectural solutions and algorithmic methods aimed at increasing the performance of computer vision systems and improving the quality and speed of visual data analysis.

The work investigates current challenges of image classification and retrieval and substantiates the need for optimization techniques in processing large-scale visual datasets. Modern approaches to neural network architectures, including convolutional networks, transfer learning, and metric learning, are analyzed. The study examines methods for generating compact embeddings, structuring the feature space, and applying optimization techniques such as pruning, quantization, fine-tuning, and accelerated nearest-neighbor search.

An optimized processing pipeline is proposed, integrating a pre-trained EfficientNet-B0 model, feature normalization, and multi-stage parameter optimization. Metric learning techniques are implemented to enhance intra-class compactness and inter-class separability, leading to improved retrieval accuracy. An experimental system was developed to perform classification, embedding generation, feature optimization, indexing of large datasets, and offline retrieval of relevant images. Testing demonstrated that the optimized approach significantly improves classification accuracy and reduces retrieval time.

The results can be applied to the development of high-performance image retrieval systems, computer vision solutions, recommendation systems, automated quality-control tools, and intelligent platforms for large-scale visual data analysis.

KEYWORDS: OPTIMIZATION OF CLASSIFICATION, MACHINE LEARNING, ARTIFICIAL I, COMPUTER VISION, NEURAL NETWORKS, IMAGE RETRIEVAL.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	11
ВСТУП.....	12
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ .....	14
1.1. Класифікація зображень .....	14
1.2. Пошук зображень за змістом .....	15
1.3. Нейронні мережі в обробці зображень .....	19
1.4. Огляд актуальних наукових робіт в галузі .....	21
2 АНАЛІЗ ТА УДОСКОНАЛЕННЯ МЕТОДІВ КЛАСИФІКАЦІЇ ПОШУКУ ЗОБРАЖЕНЬ .....	27
2.1 Аналіз існуючих методів оптимізації .....	27
2.1.1 Класичні методи класифікації та пошуку зображень .....	27
2.1.2. Методи класифікації та пошуку зображень на основі глибокого навчання .....	30
2.1.3. Методи оптимізації продуктивності та точності.....	32
2.2. Аналіз переваг та недоліків методів оптимізації класифікації та пошуку зображень .....	36
2.3 Математична модель процесу оптимізації .....	39
2.3.1 Оптимізація простору ознак за допомогою методів метричного навчання .....	44
3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ОПТИМІЗОВАНОЇ СИСТЕМИ КЛАСИФІКАЦІЇ ТА ПОШУКУ ЗОБРАЖЕНЬ ЦИФРОВИХ ОБ'ЄКТІВ.....	49
3.1 Опис розробки методу оптимізації .....	49
3.2 Використані програмні засоби.....	64
3.3 Реалізація оптимізованої моделі класифікації та пошуку зображень .....	77
ВИСНОВКИ.....	81
ПЕРЕЛІК ПОСИЛАНЬ .....	84
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....	87
ДОДАТОК Б. ЛІСТИНГ ОСНОВНИХ ПРОГРАМНИХ МОДУЛІВ.....	93

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

NN - Neural Network(нейронна мережа).

DNN - Deep Neural Network (глибока нейронна мережа).

CNN (Convolutional Neural Network) - згорткова нейронна мережа.

ResNet – Residual Network, (мережа з резидуальними блоками).

ViT - Vision Transformer (трансформер для обробки зображень).

k-NN - k-Nearest Neighbors – (метод k найближчих сусідів).

ANN - Approximate Nearest Neighbors (наближений пошук найближчих сусідів).

FAISS - Facebook AI Similarity Search (бібліотека швидкого пошуку у високовимірних просторах).

## ВСТУП

**Актуальність дослідження** зумовлена значним дисбалансом між швидкістю накопичення візуальної інформації та можливостями її структурованого аналізу. Існуючі методи обробки зображень часто демонструють або високу точність при значних обчислювальних витратах, або швидкість роботи за рахунок якості результатів. Вирішення цієї проблеми шляхом оптимізації процесів класифікації та пошуку відкриває нові перспективи для впровадження технологій машинного навчання в реальні системи.

**Наукова новизна** дослідження полягає в розробці комплексного підходу до оптимізації архітектури нейронних мереж, що враховує взаємозв'язок між точністю класифікації, швидкістю пошуку та обчислювальною складністю алгоритмів.

**Мета роботи** - Оптимізація процесу класифікації та пошуку зображень цифрових об'єктів шляхом удосконалення простору ознак та застосування методів машинного навчання, що забезпечують підвищення точності й зменшення обчислювальних витрат

**Об'єкт дослідження** - технологічний процес автоматизованої розподілу та семантичного пошуку серед масивів цифрових зображень з використанням алгоритмів машинного навчання.

**Предмет дослідження** - архітектурні рішення та алгоритмічні методи підвищення продуктивності систем комп'ютерного зору, спрямовані на вдосконалення якості та швидкості аналізу візуальних даних.

Для досягнення поставленої мети в роботі було вирішено наступні ключові завдання:

1. Аналітичний огляд існуючих рішень. Проведено комплексне дослідження сучасних наукових публікацій для систематизації методів оптимізації класифікації зображень та пошуку за змістом.

2. Експериментальна перевірка методів. Виконано практичну перевірку теоретичних підходів на репрезентативних наборах даних. Шляхом серії експериментів проведено оцінку ефективності різних алгоритмів оптимізації.
3. Розробка оптимізаційних моделей. Створено теоретичні моделі оптимізації процесів класифікації з урахуванням специфіки роботи з візуальними даними.
4. Аналіз ефективності рішень. Застосовано статистичні методи для порівняння результатів роботи різних підходів до оптимізації. Кількісно оцінено вплив оптимізаційних процедур на точність та швидкість роботи систем.
5. Порівняльна характеристика методів. Виконано комплексне порівняння ефективності різних стратегій оптимізації з використанням спеціально обраних метрик якості.
6. Удосконалення обраного підходу. Спираючись на результати проведених досліджень, розроблено покращений метод оптимізації, спрямований на підсилення ефективності алгоритмів класифікації та пошуку зображень.

Реалізація зазначених завдань забезпечила ґрунтовну основу для теоретичного дослідження та вдосконалення сучасних підходів до оптимізації процесів класифікації та пошуку зображень.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1. Класифікація зображень

Класифікація зображень (див.рис. 1.1) - це процес автоматичного віднесення цифрових зображень до певних категорій або класів на основі їхнього візуального вмісту. Основна мета класифікації полягає у визначенні тематичної приналежності зображення шляхом аналізу його характеристик та ознак. Це фундаментальна задача в таких галузях як комп'ютерний зір, системами управління цифровими колекціями, медична діагностика, автономні системи та електронна комерція.

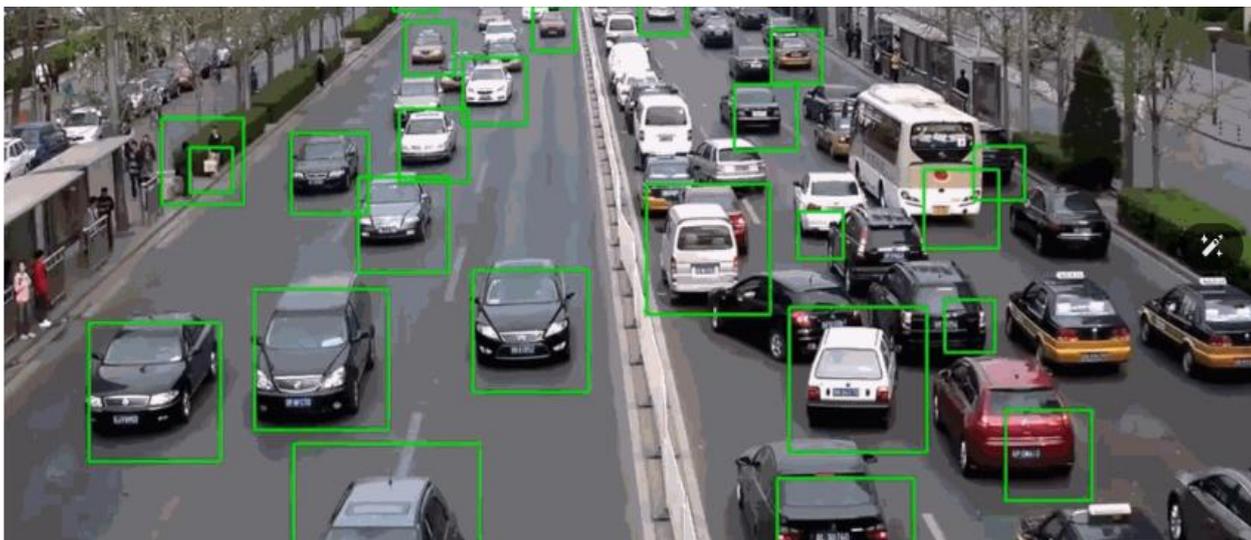


Рис. 1.1 Приклад класифікації зображень у реальному часі

Основні етапи процесу класифікації зображень охоплюють попередню підготовку вхідних даних, виявлення та виокремлення інформативних ознак, вибір і навчання відповідної моделі класифікації, а також подальшу перевірку точності отриманих результатів. Кожен із цих кроків потребує продуманої оптимізації, оскільки саме вона визначає загальну продуктивність та ефективність системи.

Техніки класифікації можуть бути різноманітними (див.рис. 1.2), включаючи традиційні методи машинного навчання, глибоке навчання на основі

згорткових нейронних мереж, а також гібридні підходи. Класифікація може бути застосована для зображень різного типу та тематики, але вимагає адаптації до таких труднощів як варіації освітлення, масштабу, ракурсу, наявності шуму та семантичної багатозначності.

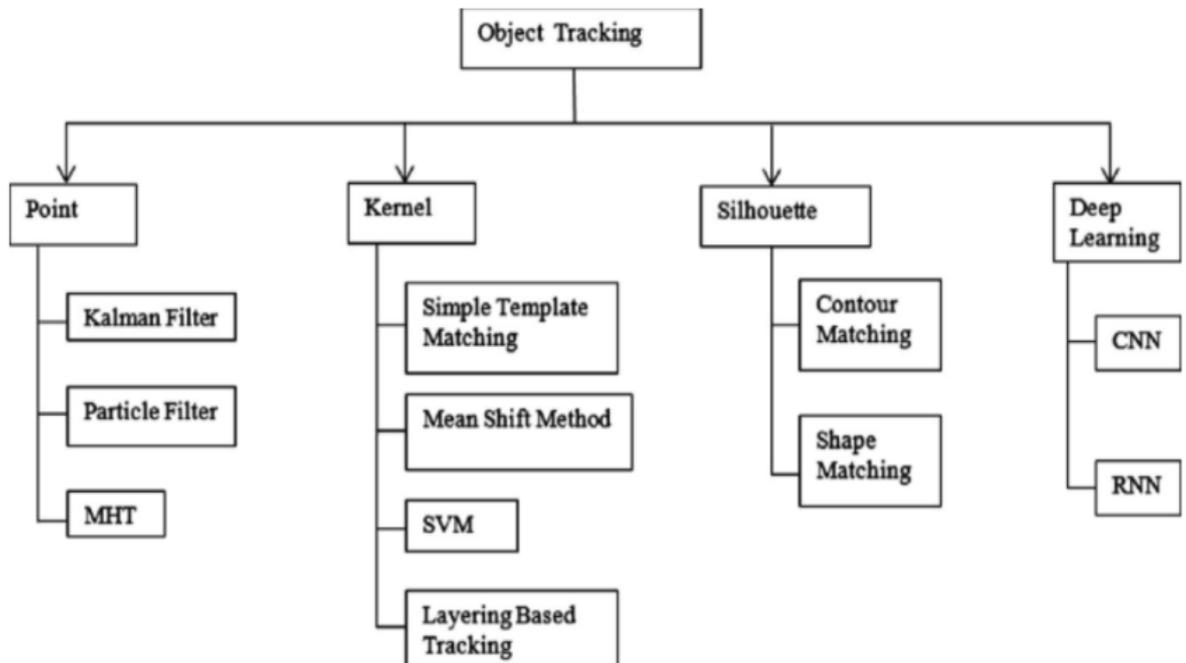


Рис. 1.2 Підходи до класифікації зображень

Використання систем класифікації зображень стає дедалі популярнішим у багатьох сферах, зокрема в системах автоматичного сортування фотографій, пошуку товарів за зображенням, медичній візуалізації, де важливо точно ідентифікувати патології, та в системах безпеки, де необхідно швидко класифікувати об'єкти для прийняття рішень.

## 1.2. Пошук зображень за змістом

Пошук зображень за змістом (див.рис. 1.3) є технологією, що базується на аналізі візуального вмісту зображень для їх автоматичної ідентифікації та пошуку. Цей підхід забезпечує результативну обробку масштабних цифрових колекціями, оскільки пошук здійснюється за візуальними ознаками, а не текстовими описами.

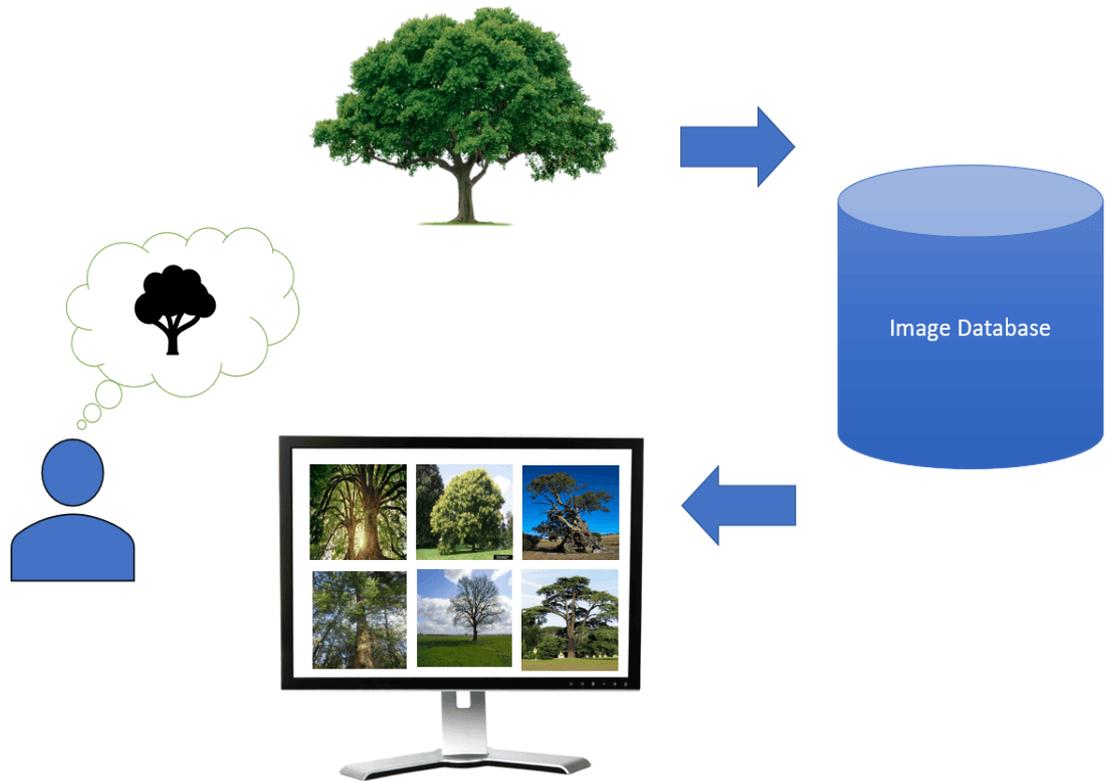


Рис. 1.3 Візуалізація роботи системи пошуку зображень змістом

Принцип роботи систем СВІР полягає в автоматичному вилученні характерних ознак зображення-запиту та їх подальшому порівнянні з ознаками зображень у базі даних. Основними типами використовуваних ознак є кольорові, текстурні, формні та просторові характеристики, які об'єднуються для формування унікального "візуального відбитка" кожного зображення.

Дана технологія знаходить широке застосування в різних сферах, наприклад:

1. Цифрові бібліотеки та архіви. Дозволяє швидко знаходити схожі зображення серед мільйонів файлів, автоматизувати процес каталогізації та полегшувати управління великими колекціями.
2. Електронна комерція. Використовується для пошуку товарів за зображенням, коли користувач може знайти аналогічні продукти, завантаживши фото товару, що його цікавить.

3. Медична діагностика. Допомагає лікарям знаходити схожі медичні зображення для порівняльного аналізу та підтримки прийняття діагностичних рішень.
4. Системи безпеки. (див.рис. 1.4) Застосовується для ідентифікації осіб, транспортних засобів або об'єктів по відеопотоку або базах даних зображень.

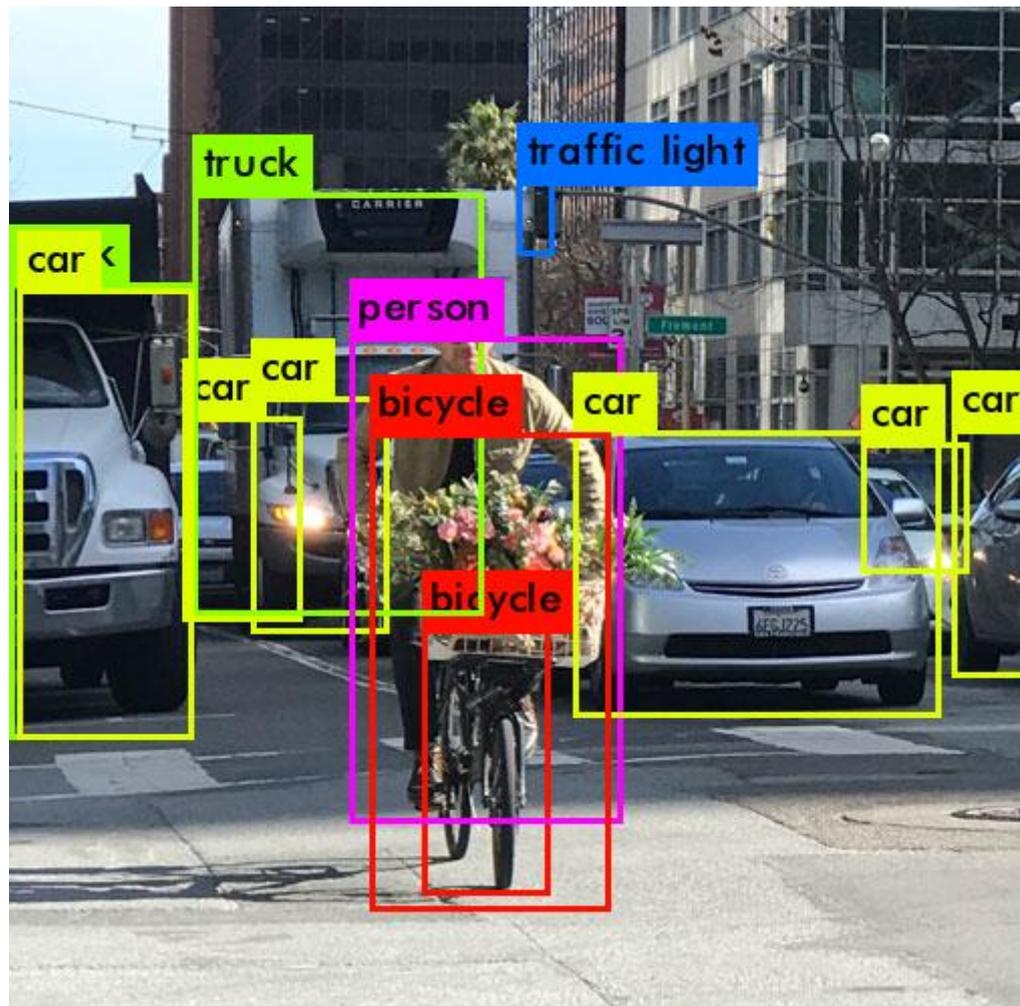


Рис 1.4 Приклад пошуку схожих зображень у системі електронної безпеки

Оптимізація систем пошуку зображень є критично важливою для підвищення їх ефективності. Основними напрямками оптимізації є:

Оптимізація вилучення ознак:

- Вибір найбільш інформативних дескрипторів
- Зменшення розмірності простору ознак
- Прискорення процесу обчислення характеристик

Оптимізація індексації та пошуку:

- Побудова ефективних структур індексів
- Використання метрик подібності, що швидко обчислюються
- Застосування методів наближеного пошуку для великих баз дани

Однією з ключових проблем у системах СВІР є семантична пропасть - розрив між низькорівневими ознаками, що витягуються комп'ютером, і високорівневими поняттями, які сприймає людина. Для подолання цієї проблеми у сучасних системах застосовуються алгоритми машинного навчання та моделі глибинних нейронних мереж.

У практичних реалізаціях, пошук зображень за змістом стає потужним інструментом для роботи з великими колекціями візуальних даних, забезпечуючи точність пошуку від 75% до 95% залежно від складності завдання та якості оптимізації алгоритмів.

Технологія СВІР особливо ефективна в таких сферах застосування:

Медіа та розваги:

- Пошук схожих зображень для контент-менеджерів
- Автоматичний розподіл фотобанків
- Модерація користувальницького контенту

Наука та дослідження:

- Аналіз супутникових знімків
- Обробка мікроскопічних зображень
- Дослідження у галузі біології та метеорології

Особисті колекції:

- Організація особистих фотографій
- Пошук дублікатів та схожих зображень
- Автоматичне тегування фотоархівів

Оптимізація систем пошуку зображень за змістом продовжує залишатися активною областю досліджень, зокрема в напрямках підвищення швидкості роботи, точності відповідностей та зменшення обчислювальних витрат.

### 1.3. Нейронні мережі в обробці зображень

Нейронні мережі є обчислювальними архітектурами, що імітують принципи роботи біологічних нейронних систем (див. рис. 1.5). В контексті обробки зображень вони складаються з взаємопов'язаних обчислювальних одиниць, організованих у шари, які спільно трансформують вхідні візуальні дані у складні абстрактні представлення, придатні для класифікації та пошуку.

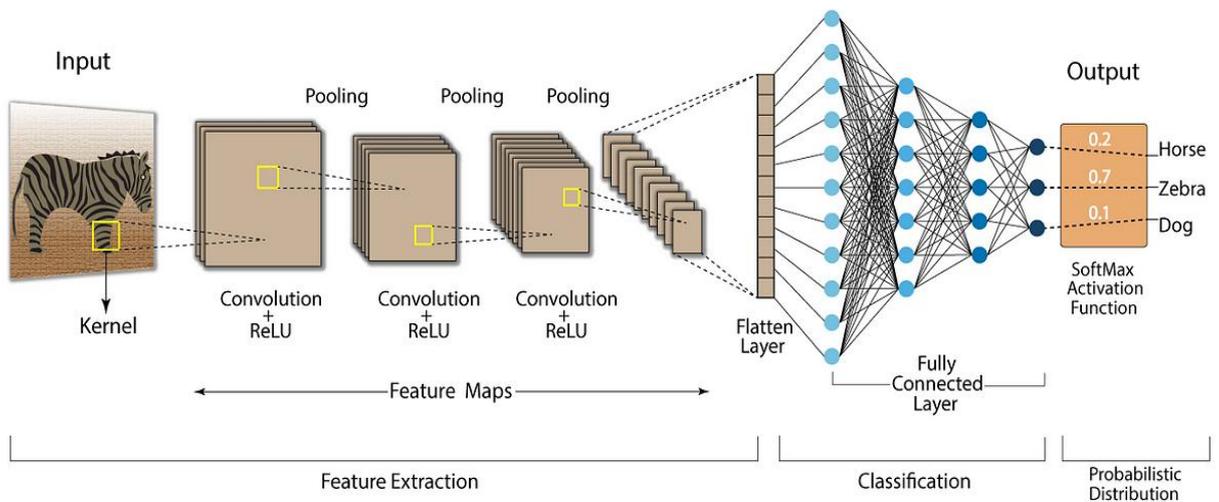


Рис. 1.5 Основна архітектура нейронної мережі, призначеної для обробки зображень

1. Архітектурні шари: Спеціалізовані шари для роботи з зображеннями містять згорткові шари, що відповідають за вилучення локальних ознак, пулінгові шари, які забезпечують зменшення розмірності даних, а також повнозв'язні шари для класифікації.
2. Параметри навчання: Вагові матриці та зсуви, що автоматично налаштовуються під час навчання для мінімізації розбіжностей між прогнозованими та реальними мітками зображень.
3. Функції активації: Нелінійні перетворення, що забезпечують здатність мережі апроксимувати складні залежності між пікселями зображення та його семантичним змістом.

4. Механізми оптимізації: Алгоритми зворотного поширення помилки разом із методами стохастичного градієнтного спуску для ефективного оновлення параметрів мережі (див. рис.1.6).

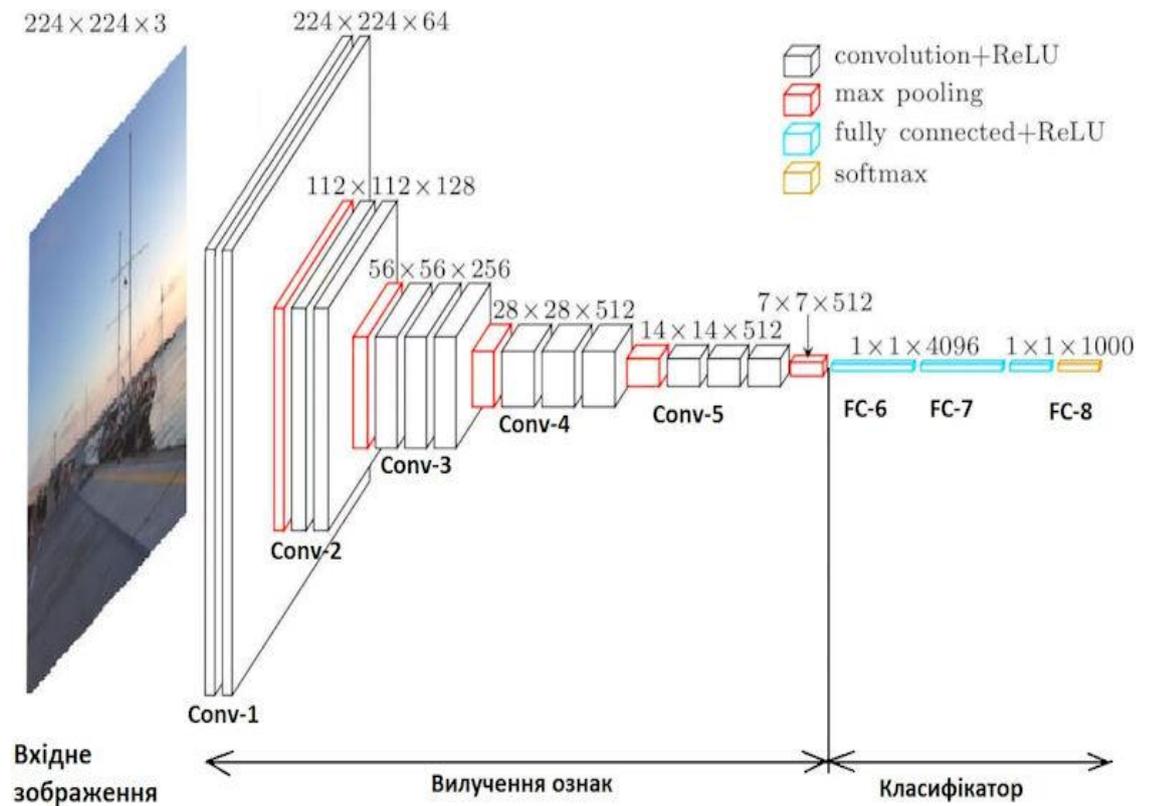


Рис. 1.6 Глибока згорткова нейронна мережа для аналізу зображень

Специфіка застосування для класифікації та пошуку зображень:

Згорткові нейронні мережі є фактичним стандартом у задачах обробки зображень, оскільки здатні автоматично формувати ієрархічні представлення ознак - від простих країв і текстур до складних об'єктів та сцен.

Оптимізаційні аспекти нейромережових архітектур для роботи з зображеннями включають:

- Скорочення надмірної параметризації через методи регуляризації
- Прискорення збіжності за допомогою нормалізації шарів
- Підвищення узагальнюючої здатності за рахунок методів аугментації даних

- Ефективне використання обчислювальних ресурсів через оптимізацію архітектури

Сучасні нейромеревеві підходи демонструють надзвичайну ефективність у таких завданнях як точна класифікація зображень за тисячами розподілу, семантичний пошук за візуальною схожістю, а також генерація та трансформація візуального контенту.

#### **1.4. Огляд актуальних наукових робіт в галузі**

Наукова праця Kantardzic M. «Deep Learning Applications» [1] представляє огляд найважливіших сфер застосування технологій глибокого навчання - починаючи від медицини до промислової автоматизації.

Основні переваги: книга охоплює практичні аспекти реалізації моделей глибокого навчання, демонструючи їх високу результативність під час розв'язання завдань розпізнавання образів, прогнозування та класифікації даних.

Актуальність роботи: у контексті становлення систем комп'ютерного зору та штучного інтелекту робить ця праця дає розуміння, як нейронні мережі інтегруються у реальні технологічні процеси.

Надзвичайно важливо: автор приділяє увагу не лише успіхам, а й обмеженням методів deep learning, що робить джерело цінним для дослідження шляхів оптимізації моделей.

Наукова робота «Object Detection in 20 Years: A Survey. International Journal of Computer Vision, Springer.» та від авторів Zou Z, Shi Z [2] аналізує еволюцію алгоритмів детектування об'єктів протягом останніх двох десятиліть.

Основні переваги: автори систематизують розвиток від класичних методів (HOG, SVM) до глибоких моделей на основі CNN і Transformer.

Актуальність роботи: дослідження дозволяє зрозуміти ключові тенденції та етапи переходу до сучасних мереж, що критично важливо для оптимізації моделей.

Надзвичайно важливо: робота підкреслює вплив зростання обчислювальних потужностей і появи великих датасетів на розвиток об'єктного розпізнавання.

Дослідження Wang J. «Visual Object Tracking from Correlation Filter to Deep Learning». [3] присвячене переходу від традиційних методів кореляційних фільтрів до глибоких нейронних мереж у відстежуванні об'єктів.

Основні переваги: детально розглянуто порівняння продуктивності класичних та глибоких методів.

Актуальність роботи: пояснюється, як сучасні мережі дозволяють значно підвищити точність трекінгу навіть у складних умовах.

Надзвичайно важливо: робота демонструє тенденцію переходу до комплексних deep learning систем із адаптивним навчанням.

Ramakrishnan S. « Visual Object Tracking with Deep Neural Networks. - Sensors, MDPI » [4] описує сучасні методи відстеження об'єктів із використанням глибоких нейронних мереж.

Основні переваги: продемонстровано використання CNN та RNN для покращення точності та стабільності трекінгу.

Актуальність роботи: дослідження важливе для розуміння того, як глибоке навчання замінює класичні підходи в системах відеоаналітики.

Надзвичайно важливо: робота формує основу для розробки систем реального часу з мінімальною похибкою відстеження.

Оглядова стаття «Deep Learning (CNN) and Transfer Learning: A Review. - Journal of AI Research» Gupta J. та колег [5] присвячена використанню transfer learning у поєднанні з CNN.

Основні переваги: описано, як повторне використання попередньо навчених моделей підвищує ефективність навчання.

Актуальність роботи: дослідження корисне для побудови оптимізованих моделей із невеликими наборами даних.

Надзвичайно важливо: підкреслено переваги transfer learning для задач класифікації та виявлення об'єктів у нових доменах.

Монографія Xu N «Video Object Tracking». [6] систематизує підходи до відстеження об'єктів у відео.

Основні переваги: представлено алгоритми трекінгу на основі CNN, LSTM і трансформерів.

Актуальність роботи: праця актуальна завдяки зростанню попиту на аналіз відеопотоків у безпеці та транспорті.

Надзвичайно важливо: книга містить приклади реалізації систем реального часу, що дозволяє адаптувати рішення до практичних задач.

Книга «Machine Learning in Robotics with Fog/Cloud Computing and IoT» Singh K. D. та Singh P. D. [7] поєднує робототехніку, IoT та машинне навчання.

Основні переваги: пояснює, як алгоритми ML інтегруються з хмарними системами керування роботами.

Актуальність роботи: підкреслює роль розподілених обчислень у системах штучного зору для автономних пристроїв.

Надзвичайно важливо: демонструє практичну цінність поєднання ML, IoT та Fog Computing для побудови адаптивних систем.

Книга Gacovski Z. «Deep Learning Algorithms» [8] надає загальний огляд основних алгоритмів глибокого навчання.

Основні переваги: чітко структуровано опис архітектур CNN, RNN, GAN та їх застосування.

Актуальність роботи: підходить для розуміння базових принципів побудови глибоких нейронних мереж.

Надзвичайно важливо: робота може слугувати теоретичною базою для розробки оптимізованих моделей комп'ютерного зору.

Книга Marron J. S. і Dryden I. L. «Object Oriented Data Analysis» [9] присвячене аналізу структурованих даних у вигляді об'єктів.

Основні переваги: розглядає методи групування та кластеризації даних для аналізу об'єктів.

Актуальність роботи: має значення для створення структурованих наборів даних для нейронних мереж.

Надзвичайно важливо: концепція використовується для моделювання взаємодій між об'єктами в задачах розпізнавання.

Робота Hinton «Reducing the Dimensionality of Data with Neural Networks» [10] описує застосування нейронних мереж для зниження розмірності даних.

Основні переваги: уперше показано використання автоенкодерів для компресії інформації.

Актуальність роботи: фундаментальна для сучасних методів feature extraction у машинному навчанні.

Надзвичайно важливо: заклала основу для deep learning як окремої дисципліни.

Стаття LeCun, Bengio, Hinton «Deep Learning» [11] - узагальнюючий огляд усіх напрямів deep learning.

Основні переваги: охоплює архітектури CNN, RNN, LSTM, Autoencoder.

Актуальність роботи: це базовий науковий фундамент, що визначив розвиток сучасного штучного інтелекту.

Надзвичайно важливо: використовується як головне джерело при вивченні основ нейронних мереж.

Roth D. E. «Real-Time Multi-Object Tracking: Methods and Applications. - IEEE Access» [12] фокусується на методах відстеження декількох об'єктів у реальному часі.

Основні переваги: пропонує аналіз сучасних алгоритмів оптимізації трекінгу (SORT, DeepSORT, FairMOT).

Актуальність роботи: важлива для систем моніторингу, безпеки та автономного транспорту.

Надзвичайно важливо: поєднання швидкодії та точності робить підхід Roth актуальним для практичних систем комп'ютерного зору.

Наукова публікація Ganmati A «Optimizing Convolutional Neural Networks for Image Classification Using Transfer Learning and Pruning Techniques» [13] розглядає сучасні методи оптимізації згорткових нейронних мереж для задач класифікації зображень.

Основні переваги: описано підхід до скорочення обчислювальної складності моделей за допомогою transfer learning та network pruning.

Актуальність роботи: у світі, де великі моделі потребують значних ресурсів, такі методи дають змогу зберігати точність при суттєвому зниженні енергоспоживання.

Надзвичайно важливо: робота показує, що ефективна оптимізація можлива навіть без втрати якості класифікації - це прямо співзвучно твоїй темі.

Trigka M., Dritsas E. Deep Learning Architectures and Optimization Techniques for Image Processing Tasks: A Comprehensive Review [14]

Основні переваги: автори систематизували методи оптимізації в CNN і Transformer-моделях, що дозволяють зменшити витрати пам'яті та час навчання.

Актуальність роботи: дослідження формує базу для вибору ефективних архітектур у завданнях обробки зображень.

Надзвичайно важливо: особлива увага приділена адаптивним оптимізаторам (Adam, RMSProp, Adagrad) та автоматичному пошуку гіперпараметрів.

Огляд Tapkir A. «An Overview of Gradient Descent Optimization Algorithms» [15] узагальнює найпопулярніші алгоритми оптимізації на основі градієнтного спуску.

Основні переваги: детально пояснено різницю між SGD, Adam, RMSProp, Adagrad та їх вплив на збіжність.

Актуальність роботи: важливе джерело для розуміння, як правильно обрати оптимізатор під конкретну архітектуру.

Надзвичайно важливо: на основі цієї роботи можна аргументувати вибір оптимізатора у твоїй дипломній як частину процесу «оптимізації класифікації».

Han S., Mao H., Dally W. J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. - ICLR, 2016.

Робота Han S. «Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding» [16] присвячена методам стиснення та оптимізації глибоких мереж.

Основні переваги: автор показують, як скоротити розмір нейронної мережі у 10–50 разів без втрати точності.

Актуальність роботи: це дозволяє використовувати складні моделі навіть на малопотужних пристроях.

Надзвичайно важливо: Deep Compression відкрила шлях до створення швидких і компактних систем класифікації зображень.

## 2 АНАЛІЗ ТА УДОСКОНАЛЕННЯ МЕТОДІВ КЛАСИФІКАЦІЇ ПОШУКУ ЗОБРАЖЕНЬ

### 2.1 Аналіз існуючих методів оптимізації

#### 2.1.1 Класичні методи класифікації та пошуку зображень

Алгоритм Viola-Jones (див. рис. 2.1), хоч і розроблений для детектування об'єктів, став важливим кроком у розвитку методів комп'ютерного зору та демонструє ключові принципи, релевантні для оптимізації. Він базується на використанні ознак Хаара та каскадних класифікаторів для швидкого відсіювання непридатних областей зображення[18].

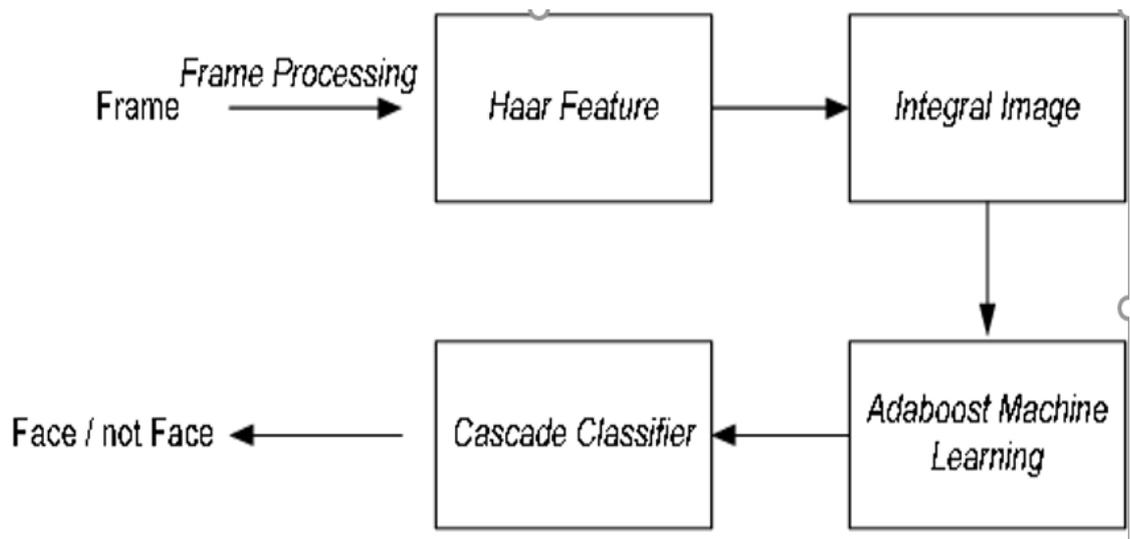


Рис. 2.1 Проста модель алгоритму Viola-Jones

Його основний внесок з позиції оптимізаційного підходу полягає в каскадному підході. Система швидко перевіряє прості ознаки на початкових етапах, відкидаючи явно невідповідні регіони, і застосовує складніші обчислення лише до перспективних ділянок. Цей принцип "відсіювання" може бути адаптований для оптимізації процесу пошуку серед великих баз зображень, де

спочатку відбираються кандидати за простими ознаками, а потім проводиться детальний аналіз за складнішими дескрипторами.

Математично, ознаки Хаара вимірюють різницю в яскравості між сусідніми прямокутними областями. Для зображення  $I(x,y)$  ознака Хаара  $H(x,y)$  обчислюється за формулою 2.1:

$$H(x, y) = \sum_{(x,y) \in A} I(x, y) - \sum_{(x,y) \in B} I(x, y) \quad (2.1)$$

де  $A$  та  $B$  - дві суміжні прямокутні області. Цей підхід дозволяє швидко виявляти контрастні структури, такі як краї.

Технічні аспекти та оптимізація:

- Швидкодія: Використання інтегральних зображень дозволяє обчислювати ознаки Хаара за константний час, що є ключем до високої швидкодії.
- Обмеження: Низька стійкість до змін масштабу, обертання та складного фону. В контексті сучасної класифікації це підкреслює важливість інваріантних ознак.
- Урок для оптимізації: Алгоритм демонструє, що ефективність часто досягається не за рахунок складності окремих операцій, а завдяки розумній архітектурі процесу обробки, що мінімізує загальний обсяг обчислень.

Histogram of Oriented Gradients є класичним методом вилучення ознак, який значно ближчий до задач класифікації та пошуку. Він описує структуру об'єкта за допомогою розподілу напрямків градієнтів яскравості, формуючи стійкий до змін освітлення дескриптор.

Метод ділить зображення на невеликі комірки, для кожної з яких будується гістограма орієнтацій градієнтів. Сукупність цих гістограм утворює детальний опис форми та контурів об'єкта.

Математична основа HOG[19]:

Для кожного пікселя обчислюються за формулою горизонтальний  $G_x$  (2.2) та вертикальний  $G_y$  (2.3) градієнти:

$$G_x = I(x + 1, y) - I(x - 1, y) \quad (2.2)$$

$$G_y = I(x, y + 1) - I(x, y - 1) \quad (2.3)$$

За ними визначаються магнітуда (2.4) та орієнтація градієнта (2.5) обчислюються за формулою:

$$Magnitude = \sqrt{G_x^2 + G_y^2} \quad (2.4)$$

$$Orientation = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.5)$$

Оптимізаційний потенціал та недоліки:

- Переваги: Висока стійкість до незначних змін освітлення та геометричних спотворень порівняно з піксельними методами.
- Недоліки: Фіксований розмір дескриптора, відсутність інваріантності до масштабу та обертання, обмежена здатність описувати глобальний контекст зображення.
- Зв'язок з сучасними методами: HOG можна розглядати як попередника згорткових шарів у нейронних мережах. Аналіз його обмежень допомагає зрозуміти, чому глибоке навчання, яке самостійно навчає ієрархічні ознаки, стало настільки ефективним. Для оптимізації сучасних систем HOG може використовуватись на початкових стадіях, коли швидкий фільтр для попереднього відбору кандидатів.

## 2.1.2. Методи класифікації та пошуку зображень на основі глибокого навчання

Методи на основі глибокого навчання (див. рис. 2.2) здійснили революцію в задачах класифікації та пошуку зображень, забезпечивши значне підвищення точності завдяки здатності автоматично виявляти складні ієрархічні ознаки. Ці підходи базуються на застосуванні глибоких нейронних мереж для опрацювання візуального вмісту.

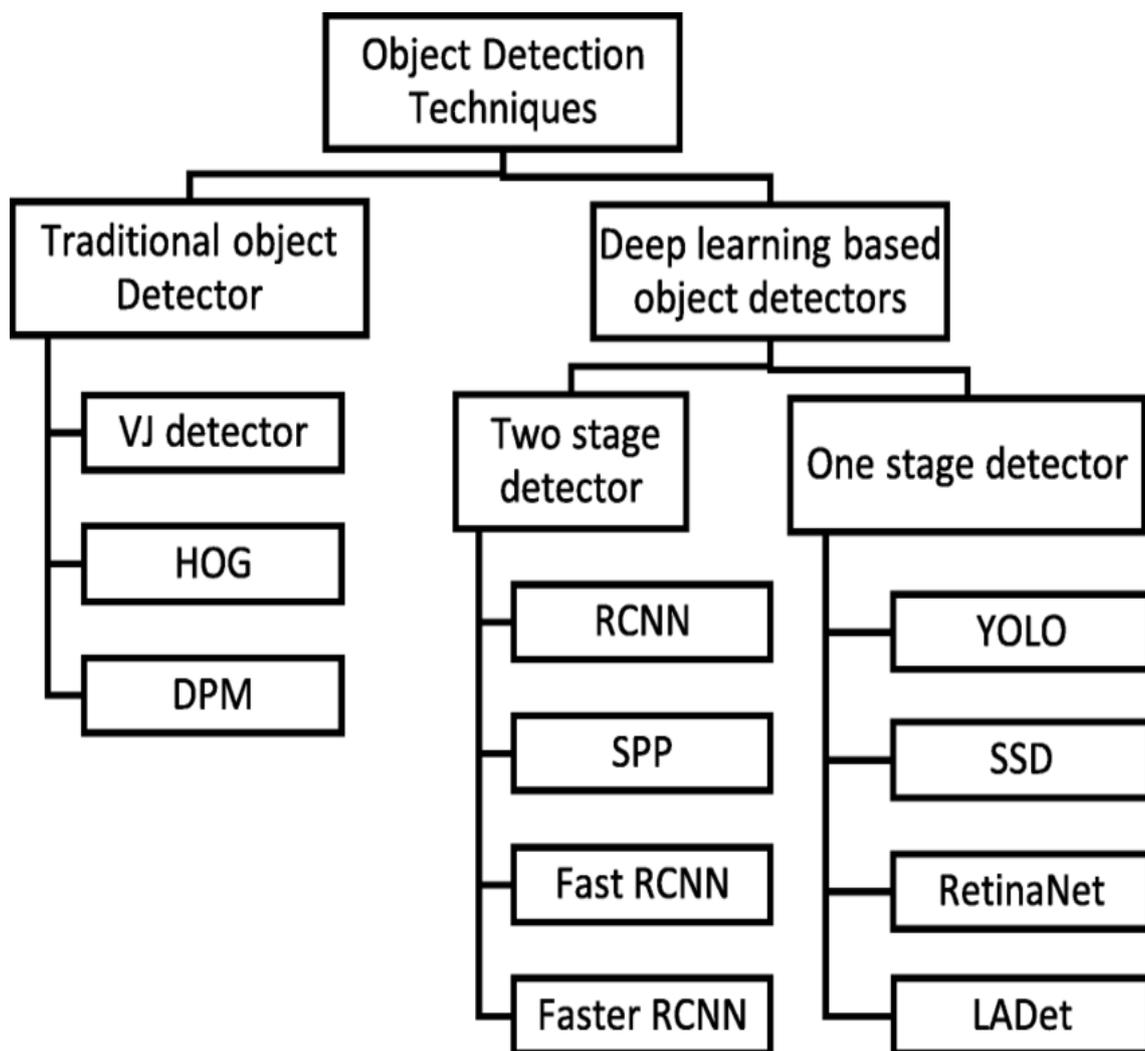


Рис. 2.2 Базові архітектури глибокого навчання для аналізу зображень

Згорткові нейронні мережі є фундаментом для більшості сучасних методів. Їхня архітектура, що імітує зорову кору, дозволяє ефективно виявляти просторові залежності в зображеннях. Ключовою перевагою CNN є здатність до

автоматичного навчання ознак, що усуває необхідність ручного конструювання дескрипторів, як у класичних методах[17].

Архітектури-класифікатори, такі як VGG[20], ResNet[21], EfficientNet[22] та Vision Transformers[14], використовуються для завдання розподілу зображень. Вони працюють за принципом від загального до часткового:

- Ранні шари виявляють прості ознаки.
- Глибокі шари формують абстрактні поняття.

Для задач пошуку за змістом ці мережі використовуються як потужні "витягувачі ознак". Вектор ознак, отриманий з одного з останніх шарів мережі, використовується як унікальний "відбиток" зображення. Пошук здійснюється шляхом порівняння цих векторів за допомогою метрик подібності (наприклад, косинусної подібності чи евклідової відстані).

Математична основа CNN:

Основним будівельним блоком CNN є згорткова операція, яка застосовує набір фільтрів (ядер) до вхідного образу з метою виявлення локальних ознак обчислюється за формулою (2.6):

$$S(i, j) = (I \times K)(i, j) = \sum_m \sum_n I(m, n) \times K(i - m, j - n) \quad (2.6)$$

де:

$S(i, j)$ - активація вихідної карти ознак у позиції  $(i, j)$ ;

$I$ - вхідне зображення або карта ознак з попереднього шару;

$K$ - ядро згортки (фільтр).

Шари пулінгу, такі як максимальне пулінгу, зменшують розмірність даних, зберігаючи найважливіші ознаки та збільшуючи інваріантність до невеликих зсувів обчислюється за формулою:

$$M(i, j) = \max(I(i, j)) \quad (2.7)$$

де  $M(i, j)$  - вихідне значення після операції пулінгу.

Аспекти оптимізації сучасних методів:

- **Трансферне навчання:** Дозволяє використовувати попередньо навчені на масштабних вибірках даних, (таких як ImageNet) моделі як стартову точку для специфічних задач. Це значно скорочує час навчання, вимоги до обчислювальних ресурсів та обсягу розмічених даних, що є ключовою оптимізацією в практичних умовах.
- **Порівняння архітектур:** Різні архітектури пропонують різний баланс між точністю та обчислювальною ефективністю. Наприклад, EfficientNet спеціально оптимізований для досягнення максимальної точності при мінімальній кількості параметрів, що робить його ідеальним кандидатом для впровадження в системах з обмеженими ресурсами.
- **Метричне навчання (Metric Learning)[29]:** Спеціалізовані підходи, такі як використання контрастивних або triplet-функцій втрат, напряду оптимізують простір ознак для пошуку. Вони "навчають" мережу таким чином, що семантично схожі зображення проектуються близько одна до одної в просторі ознак, а несхожі - далеко. Це дозволяє подолати "семантичну прірву" ефективніше, ніж використання звичайних класифікаційних мереж.

### **2.1.3. Методи оптимізації продуктивності та точності**

Підвищення ефективності систем комп'ютерного зору є критично важливим для їх практичного впровадження. Сучасні методи оптимізації можна класифікувати на кілька основних категорій, спрямованих на прискорення роботи, зменшення споживання ресурсів та підвищення точності.

#### **1. Оптимізація архітектури моделей**

Цей підхід спрямований на зменшення розміру та складності нейронної мережі без суттєвої втрати точності (див. рис. 2.3).

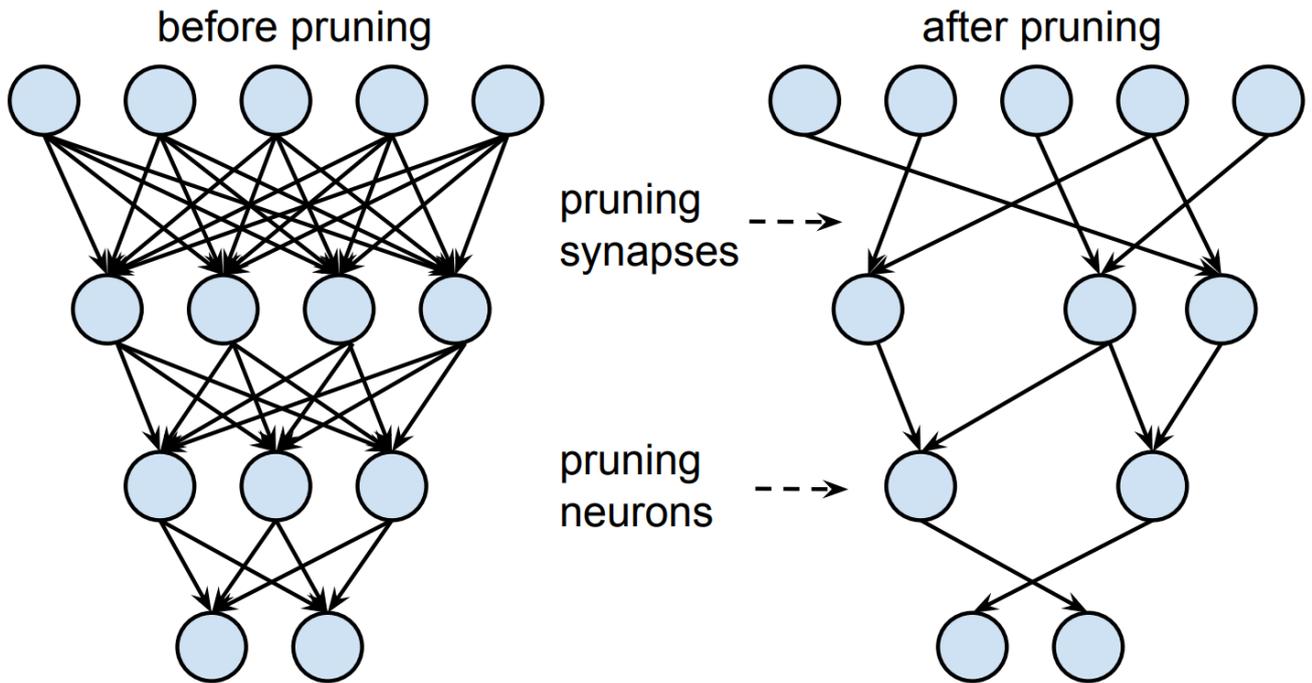


Рис. 2.3 Принцип оптимізації моделі шляхом видалення маловажливих зв'язків

До ключових технік належать:

- Обрізання (Pruning)[16]: Видалення маловажливих ваг, нейронів або цілих шарів мережі, що призводить до зменшення розміру моделі та прискорення її роботи.
- Квантування (Quantization)[16]: Зменшення розрядності числових представлень ваг мережі (наприклад, з 32-бітних чисел з рухомою комою до 8-бітних цілих чисел). Це забезпечує пряме зменшення обсягу пам'яті та прискорення обчислень на підтримуваному апаратному забезпеченні.
- Дистиляція знань (Knowledge Distillation)[17]: Техніка, коли мала, "учнівська" мережа навчається імітувати поведінку великої, складеної "вчительської" мережі. В результаті невелика модель досягає точності, близької до оригіналу, але працює значно швидше.

## 2. Оптимізація процесу навчання

Ці методи дозволяють прискорити збіжність моделі та покращити[23] її узагальнюючу здатність (див. рис. 2.4).

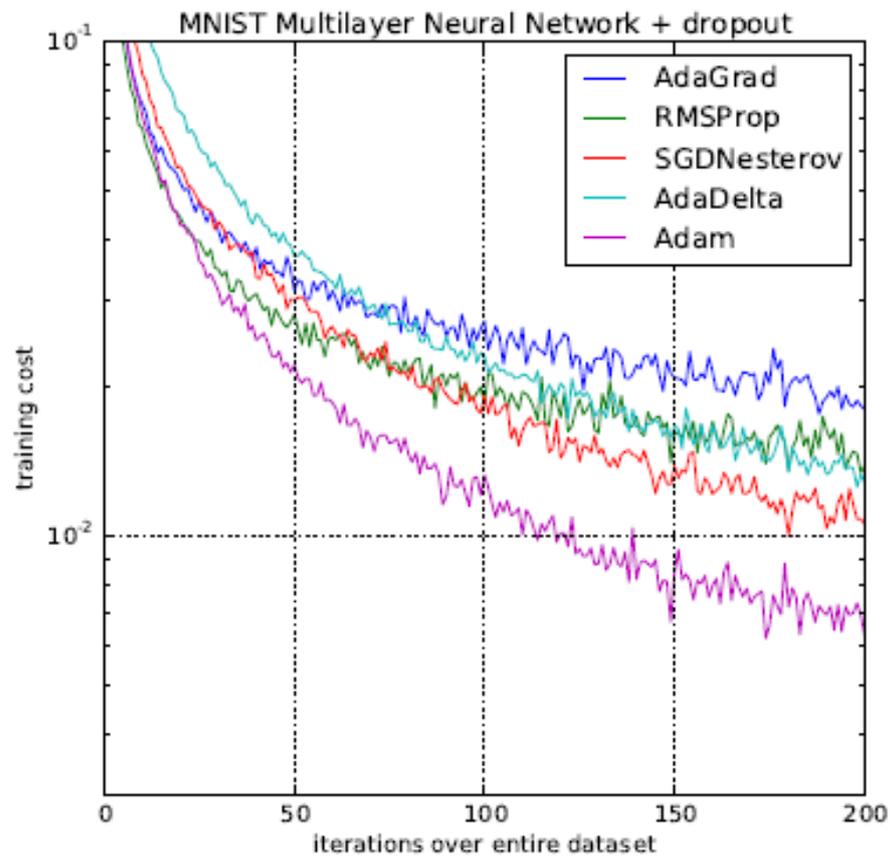


Рис. 2.4 Вплив різних оптимізаторів на швидкість збіжності моделі[23]

Основні напрямки:

- Адаптивні оптимізатори: Використання алгоритмів на кшталт Adam, RMSProp, які автоматично налаштують темп оновлення вагових параметрів мережі, що забезпечує прискорену та стабільнішої збіжності порівняно зі стандартним стохастичним градієнтним спуском.
- Планувальники швидкості навчання: Стратегії динамічної зміни швидкості навчання під час тренування (наприклад, зменшення за кроками або за косинусом) для більш точного "настроювання" ваг мережі та подолання локальних мінімумів.
- Аугментація даних : Стратегічне доповнення навчального набору даних штучно створеними зразками (обертання, зміна масштабу, кольору, додавання шуму). Це покращує здатність моделі до узагальнення і запобігає перенавчанню, що є формою оптимізації її якості.

### 3. Оптимізація інфраструктури та інференсу

Цей підхід фокусується на ефективному виконанні вже навченої моделі у робочих умовах (див. рис. 2.5).

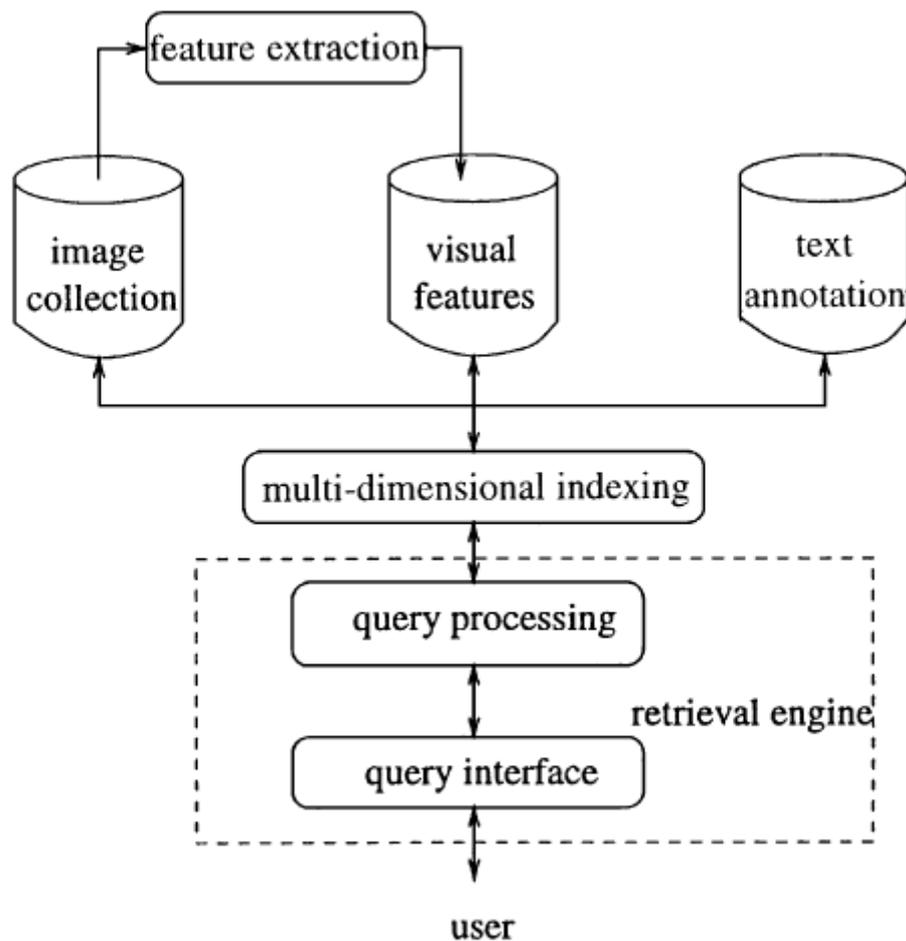


Рис. 2.5 Архітектура високопродуктивної системи пошуку зображень [27]

Сюди належать:

- Спеціалізовані апаратні прискорювачі: Використання графічних процесорів (GPU), тензорних процесорів (TPU) та спеціальних мікросхем (ASIC) для максимально швидкого виконання матричних операцій, що становлять основу роботи нейронних мереж.
- Оптимізація програмного забезпечення: Використання високоефективних бібліотек, таких як TensorFlow Lite, ONNX Runtime або NVIDIA TensorRT, які застосовують низькорівневі оптимізації

(оптимізація графу обчислень, кешування тощо) для прискорення інференсу.

- Ефективна індексація для пошуку: Для завдань пошуку серед мільйонів зображень використовуються спеціалізовані структури даних та алгоритми наближеного пошуку найближчих сусідів (Approximate Nearest Neighbors, ANN), такі як FAISS або HNSW. Вони дозволяють значно скоротити час пошуку при прийнятному рівні точності.

## **2.2. Аналіз переваг та недоліків методів оптимізації класифікації та пошуку зображень**

Сучасні методи оптимізації на основі глибокого навчання дозволяють досягати високої ефективності систем комп'ютерного зору у різних умовах експлуатації. Наприклад, оптимізовані архітектури нейронних мереж забезпечують точність класифікації понад 95% на складних наборах даних, таких як ImageNet, при значному скороченні часу обробки.

### **Переваги сучасних методів оптимізації**

Глибоке навчання забезпечує автоматичне виявлення оптимальних ознак зображень без необхідності ручного проектування характеристик. Це дозволяє створювати універсальні системи, можуть результативно функціонувати з різноманітними видами зображень - від медичних знімків до супутникових фотографій. Сучасні оптимізовані архітектури, такі як EfficientNet чи MobileNet, демонструють високу продуктивність у тому числі на апаратних платформах із обмеженими обчислювальними можливостями.

Методи трансферного навчання значно скорочують вимоги до обсягів навчальних даних, дозволяючи адаптувати потужні попередньо навчені моделі для вирішення специфічних завдань. Техніки метричного навчання забезпечують точний семантичний пошук зображень, ефективно подолуючи "семантичну прірву" між низькорівневими ознаками та високорівневими поняттями.

## Недоліки та обмеження

Основним недоліком сучасних методів залишається висока обчислювальна складність процесу навчання моделей. Навіть оптимізовані архітектури вимагають потужних GPU для ефективного тренування, що може бути перешкодою для невеликих проєктів.

Іншою суттєвою проблемою є залежність якості роботи від обсягу та якості розмітки навчальних даних. Досягнення високої точності вимагає використання масштабних наборів даних з ретельними анотаціями, створення яких вимагає суттєвих витрат часу та фінансових ресурсів.

Багато сучасних методів демонструють зниження ефективності при роботі з даними, що відрізняються від навчального розподілу. Це обмежує їх застосування в динамічних середовищах, де умови можуть змінюватися.

Питання енергоефективності також залишається актуальним, особливо для систем реального часу, що працюють на мобільних пристроях або в умовах обмеженого живлення.

## Перспективи вдосконалення

Подальший розвиток методів оптимізації пов'язаний з пошуком балансу між точністю, швидкістю та енергоспоживанням. Перспективними напрямками є розробка спеціалізованих архітектур, адаптивних алгоритмів навчання та методів автоматичної оптимізації гіперпараметрів.

Впровадження цих методів у практичних системах вимагає ретельного аналізу вимог конкретного застосування та врахування обмежень доступних обчислювальних ресурсів.

Порівняння методів оптимізації класифікації та пошуку зображень зобразимо таблицю 2.1:

Таблиця 2.1

## Порівняння методів оптимізації класифікації та пошуку зображень

Характеристика	Класичні методи (HOG, SIFT)	Базові CNN (наприклад, VGG)	Оптимізовані CNN (наприклад EfficientNET)	Методи з трансферним навчанням
Точність	Середня	Висока	Дуже висока	Висока
Швидкість інтерфейсу	Висока	Середня/Низька	Висока	Середня/Висока
Обчислювальна складність	Низька	Висока	Середня	Середня
Вимоги до обсягу даних	Невеликі	Дуже великі	Великі	Невеликі
Зручність оптимізації	Низька	Середня	Висока	Невеликі/Середні
Придатність для пошуку	Обмежена	Задовільна	Висока	Висока

Оцінка ефективності методів оптимізації класифікації та пошуку зображень представлені на таблиці 2.2

Таблиця 2.2

## Оцінка ефективності методів оптимізації

Метод оптимізації	Ефективність (%)	Вплив на швидкість (%)	Складність впровадження (%)	Загальна корисність (%)
Трансферне навчання	93	60	40	95
Аугментація даних	81	80	30	90
Обрізання моделі	76	85	70	80
Квантування	60	90	60	80
Метричне навчання	90	60	80	90
Гіперпаоаметричний пошук	85	50	80	70

### 2.3 Математична модель процесу оптимізації

Математична модель у даній роботі служить формальним інструментом для опису та вирішення задачі оптимізації систем класифікації та пошуку зображень. На відміну від моделей, спрямованих на створення нових алгоритмів розпізнавання, модель зосереджена на критеріях прийняття рішень для вдосконалення вже існуючих систем.

Основне завдання оптимізації полягає у знаходженні такого стану системи, який забезпечує найкращий баланс між її основними характеристиками. У цьому випадку це точність, швидкість роботи та вимоги до ресурсів[1].

Формально цю задачу можна представити у вигляді цільової функції, яку необхідно мінімізувати або максимізувати.

### 1. Узагальнена цільова функція оптимізації

Нехай  $S$  - це система класифікації та пошуку зображень, а Конфігурація( $S$ ) - сукупність її налаштувань (архітектура нейронної мережі, гіперпараметри, методи попередньої обробки даних).

Тоді математично задачу оптимізації можна сформулювати так:

Знайти конфігурацію( $S$ ), для якої досягається мінімум цільової функції  $F(S)$  обчислюється за формулою (2.8):

$$F(S) = \alpha \times [1 - Accuracy(s)] + \beta \times Time(S) + \gamma \times Size(S) \quad (2.8)$$

де:

$Accuracy(S)$  - точність класифікації системи  $S$  (наприклад, відсоток правильно розпізнаних зображень). Чим вища точність, тим менше перший доданок  $[1 - Accuracy(S)]$ .

$Time(S)$  - середній час обробки одного зображення системою  $S$  (інференс).

$Size(S)$  - розмір моделі в пам'яті (у мегабайтах).

$\alpha$ ,  $\beta$ ,  $\gamma$  -вагові коефіцієнти, які визначають важливість кожного критерію.

Наприклад:

Якщо для нас найважливіша точність, ми встановлюємо  $\alpha$  значно більшим за  $\beta$  та  $\gamma$ .

Якщо система має працювати на мобільному пристрої, ми збільшуємо вагу  $\gamma$  (вплив розміру моделі) та  $\beta$  (вплив часу роботи).

Пояснення моделі:

Ця модель є спрощеним, але ефективним узагальненням. Вона не описує внутрішню будову нейронної мережі, а формалізує компроміс, який ми робимо при її оптимізації.

Якщо ми хочемо дуже точну систему (максимізувати Accuracy), нам, швидше за все, доведеться погодитися на велику модель Size(S) зростає та повільну обробку Time(S) зростає.

Якщо нам потрібна дуже швидка система мінімізувати Time(S), ми, ймовірно, будемо використовувати спрощену модель, що може призвести до падіння точності Accuracy(S) зменшується.

Таким чином, процес оптимізації зводиться до:

- Вимірювання поточних значень Accuracy(S), Time(S), Size(S).
- Вибору вагових коефіцієнтів  $\alpha$ ,  $\beta$ ,  $\gamma$  відповідно до цілей проекту.
- Пошуку такої Конфігурації(S), яка мінімізує значення функції F(S).

Ця модель є фундаментом для подальшого застосування конкретних оптимізаційних технік, таких як трансферне навчання, аугментація даних чи стиснення моделей, кожна з яких по-своєму впливає на компоненти цільової функції F(S).

Основні характеристики математичної моделі:

- Узагальнення реальних процесів. Математична модель подає складну реальну систему у спрощеному вигляді, зберігаючи ключові елементи, необхідні для її подальшого дослідження.
- Формалізоване подання зв'язків. Вона описує взаємодію між окремими частинами системи або процесу через математичні формули та залежності.
- Можливість прогнозування й дослідження. Такі моделі дають змогу передбачати поведінку системи, перевіряти різні сценарії та оцінювати вплив окремих чинників на її роботу.
- Раціональність і обчислювальна користь. Модель дає змогу виконувати обчислення та робити висновки на основі математичних методів і алгоритмів.

Слід пам'ятати, що математична модель - це спрощена інтерпретація реальності, яка може містити певні обмеження та припущення, і їх необхідно враховувати під час використання моделі та тлумачення отриманих результатів.

У своїй роботі я працював над оптимізацією системи, що включає як задачу класифікації, так і задачу пошуку за змістом, і відповідно використовував математичні моделі, пов'язані з цими особливостями.

Розглянемо математичні моделі та формули, які застосовуються в досліджуваній системі оптимізації:

## 2. Формалізація задачі багатокритеріальної оптимізації:

Задачу оптимізації можна представити як пошук такої конфігурації моделі  $M$  (архітектура + параметри), яка мінімізує складену цільову функцію, що враховує як якість, так і ефективність обчислюється за формулою (2.9):

$$M = \text{ArgMin}(a \times L(M) + \beta \times c(M) + \gamma \times t(M)) \quad (2.9)$$

де:

$L(M)$  - функція втрат моделі  $M$  (наприклад, categorical cross-entropy), що відображає помилку класифікації. Чим менше  $L(M)$ , тим вища точність.

$C(M)$  - функція вартості, що оцінює обчислювальну складність моделі (наприклад, кількість параметрів, FLOPs).

$T(M)$  - функція, що оцінює час інференсу моделі на цільовому пристрої.

$\alpha$ ,  $\beta$ ,  $\gamma$  - вагові коефіцієнти, що визначають пріоритет кожного критерію в загальній цільовій функції. Наприклад, для системи реального часу  $\gamma$  буде високим.

Обмеженнями можуть виступати:

- $\text{Accuracy}(M) > A_{\min}$  - мінімально прийнятна точність.
- $\text{Memory}(M) < S_{\max}$  - максимально допустимий розмір моделі в пам'яті.
- $\text{Latency}(M) < T_{\max}$  - максимально допустима затримка (час відповіді).

### 3. Модель трансферного навчання:

Трансферне навчання формалізується як задача донавчання попередньо навченої моделі  $M_{base}$  на новому наборі даних  $D_{target}$ . Нехай  $\theta_{base}$  - ваги попередньо навченої моделі, а  $\theta_{new}$  - ваги, що навчаються. Процес оптимізації можна представити як обчислюється за формулою (2.10):

$$\theta = \text{ArgMin}(L(D_{target}; \theta_{base\_frozen}, \theta_{new})) \quad (2.10)$$

де частина ваг  $\theta_{base}$  заморожується (не оновлюється під час навчання), що значно скорочує кількість параметрів, що підлягають оптимізації, та час навчання.

### 4. Модель стиснення мережі (Обрізання - Pruning)[16]:

Мета обрізання - видалити маловажливі вагові зв'язки. Введемо маску  $m \in \{0,1\}$  для кожного параметра  $w$  моделі. Оптимізовані ваги  $w$  обчислюються як обчислюється за формулою (2.11):

$$w = m \odot w \quad (2.11)$$

де  $\odot$  - поелементне множення (добуток Адамара). Завдання полягає у знаходженні такої маски  $m$ , що мінімізує зростання функції втрат  $|L(M) - L(M_{pruned})|$  при максимальному рівні розрідженості (кількість нулів в  $m$ ) / загальна кількість параметрів`.

Модель квантування:

Квантування зменшує розрядність представлення ваг. Нехай  $w_{fp32}$  - вага у форматі FP32. Процес квантування до INT8 можна описати як обчислюється за формулою (2.12):

$$w_{int8} = \text{Round}\left(\frac{w_{fp32} - offset}{scale}\right) \quad (2.12)$$

де *scale* - коефіцієнт масштабування, а *offset* - зсув. Це перетворення дозволяє зменшити розмір моделі в 4 рази та прискорити обчислення на спеціалізованому апаратному забезпеченні.

Математичні моделі для оцінки якості пошуку:

Для задачі пошуку зображень за змістом ключовим є порівняння векторних представлень. Нехай  $f(I)$  - функція, що відображає зображення  $I$  у вектор ознак у  $d$ -вимірному просторі.

- Косинусна подібність обчислюється за формулою(2.13):

$$\text{sim}(A, B) = (A \times B) / (||A|| \times ||B||) \quad (2.13)$$

де  $A$  та  $B$  - вектори ознак двох зображень. Ця метрика є інваріантною до норми вектора, що важливо для пошуку за візуальним змістом.

- Відстань L2 (Евклідова відстань) обчислюється за формулою (2.14):

$$d(A, B) = \sqrt{(\sum(A_i - B_i)^2)} \quad (2.14)$$

Чим менша відстань, тим більша подібність між зображеннями.

Для оцінки оптимізації інфраструктури пошуку використовується метрика прискорення обчислюється за формулою (2.15):

$$\text{Speedup} = T_{\text{original}} / T_{\text{optimized}} \quad (2.15)$$

де  $T_{\text{original}}$  - час пошуку в неоптимізованій системі, а  $T_{\text{optimized}}$  - час пошуку після впровадження оптимізацій.

### **2.3.1 Оптимізація простору ознак за допомогою методів метричного навчання**

Оптимізація простору ознак (feature space) є однією з головних складових процесу підвищення ефективності класифікації та пошуку цифрових зображень.

Незалежно від архітектури нейронної мережі, саме якість сформованих векторних представлень (embedding-ів) визначає, наскільки точно система зможе порівнювати об'єкти, знаходити подібні зображення та відрізнити семантично різні класи.

Проблема полягає в тому, що стандартне навчання нейромережі - навіть потужних моделей, таких як EfficientNet, ResNet чи Vision Transformers - не гарантує, що простір embedding-ів буде оптимально структурованим для подальшого пошуку.

У не оптимізованому просторі спостерігаються такі явища:

- перекриття класів - зображення різних категорій знаходяться надто близько.
- внутрішньокласова розбіжність - зображення одного класу мають велику варіативність і лежать далеко один від одного.
- відсутність чіткої геометричної структури, що заважає застосовувати методи найближчих сусідів (k-NN) та інші алгоритми пошуку.

Метричне навчання вирішує ці проблеми, змінюючи підхід до формування ознак не через максимізацію точності класифікації, а через оптимізаційні функції відстаней, які безпосередньо керують взаємним розташуванням векторів у багатовимірному просторі.

Формування інваріантного простору ознак:

Метою метричного навчання є побудова такого простору, де:

- вектори схожих об'єктів мають малу відстань;
- вектори різних об'єктів — велику;
- простір містить компактні кластери, що відповідають класам або категоріям;
- моделі структуровано розділяють семантику зображень.

Для цього використовуються спеціальні функції втрат.

Контрастивна функція втрат (Contrastive Loss)[24]

Контрастивна функція втрат оптимізує простір ознак на основі порівняння пар зображень.

Нехай маємо пару зображень  $A$  і  $B$  та андикатор подібності  $Y$ :

- $Y=0$  – позитивна пара (схожі об'єкти),
- $Y=1$  – негативна пара (різні об'єкти),

Тоді функція втрат набуває вигляду обчислюється за формулами (2.16, 2.17):

$$L_{contrastive} = (1 - Y) \frac{1}{2} d(A, B)^2 + Y \frac{1}{2} (\max(0, a - d(A, B)))^2 \quad (2.16)$$

$$d(A, B) = \|f(A) - f(B)\|_2 \quad (2.17)$$

де:

$f(\cdot)$  – вектор ознак

$a$  – відступ (margin), мінімальна очікувана відстань для різних класів.

Суть:

- якщо об'єкти схожі, модель зменшує їхню відстань:
- якщо різні – збільшує, але не безмежно, а лише до рівня  $a$ .

Таким чином формується геометрично добре структурований простір ознак.

Центроїдна функція втрат (Center Loss)[25]

Контрастивна функція не розв'язує проблему надмірної внутрішньокласової варіативності.

Для цього застосовують Center Loss, що “притягує” вектори одного класу до їх центру обчислюється за формулою (2.18):

$$L_{center} = \frac{1}{2} \sum_{i=1}^m \|f(x_i) - c_{y_i}\|^2 \quad (2.18)$$

де:

- $f(x_i)$  – embedding

- $c_{yi}$  – центроїд класу  $y_i$
- $m$  – розмір батчу

Цей метод робить усі вектори одного класу ближчими до центра:

- покращує клстеризацію,
- підвищує точність k-NN-пошуку,
- зменшує помилки класифікації у зонах перетину.

Особливість: Center Loss вимагає балансу з крос-ентропією обчислюється за формулою (2.19):

$$L = L_{softmax} + \lambda L_{center} \quad (2.19)$$

де  $\lambda$  – коефіцієнт впливу оптимізації простору ознак.

Нормалізація embedding [11]

Для стабільності пошуку та класифікації важливо, щоб всі вектори ознак мали однаковий масштаб. Тому застосовується L2-нормалізація обчислюється за формулою (2.20):

$$f(x) = \frac{f(x)}{\|f(x)\|_2} \quad (2.20)$$

Нормалізація:

- робить косинусну відстань більш інформативною,
- рівномірно розподіляє вектори на поверхні гіперсфери,
- покращує збіжність метричного навчання.

Angular Loss та оптимізація кутових відстаней

У задачах пошуку часто важлива кутова, а не евклідова подібність.

Тому застосовують функції втрат, що оптимізують саме кут між векторами обчислюється за формулою обчислюється за формулою (2.21):

$$L_{angular} = \max(0, \cos(\theta_{A,N}) - \cos(\theta_{A,P}) + a) \quad (2.21)$$

де:

- $\theta_{A,P}$  – кут між якірем і позитивним об’єктом,
- $\theta_{A,N}$  – кут між якірем і негативним.

Перевага: клас стає не просто “ближчим”, а кутово відокремленим, що покращує пошук у високих розмірностях.

Геометричне значення оптимізованого простору

Після застосування метричного навчання embedding-простір отримує такі властивості:

- внутрішньокласова компактність,
- міжкласова роздільність,
- стійкість до шуму, освітлення, ракурсу,
- монотонність метрик у відношенні семантики,
- покращення пошуку найближчих сусідів,
- ліпша кластеризація,
- вища точність класифікації навіть без зміни архітектури моделі.

## **3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ОПТИМІЗОВАНОЇ СИСТЕМИ КЛАСИФІКАЦІЇ ТА ПОШУКУ ЗОБРАЖЕНЬ ЦИФРОВИХ ОБ'ЄКТІВ**

### **3.1 Опис розробки методу оптимізації**

Завдання оптимізації пошуку й класифікації цифрових зображень є одним із ключових напрямів сучасного комп'ютерного зору. Незважаючи на значний прогрес у глибинному навчанні, проблема ефективного пошуку подібних зображень у великих колекціях залишається складною, особливо коли об'єкти різняться за видом, масштабом, ракурсом, фоном або умовами зйомки. У межах цієї роботи розглядається розробка оптимізованої системи для класифікації та пошуку зображень, що дозволяє визначати зображення потрібного класу - у даному випадку зображень котів - серед великого набору фотографій, які включають не лише зображень котів, а й зображень собак і зображень папуг. Цей приклад обрано через його універсальність: собаки та коти мають схожі кольорові й текстурні ознаки, а папуги часто демонструють виражені особливості форми та кольору, що створює широкий діапазон складності для тестування алгоритмів.

На відміну від класичних підходів, що зосереджені лише на класифікації за попередньо визначеними ознаками, сучасні системи повинні забезпечувати обидві функції - як точну класифікацію, так і високоточний пошук релевантних зображень на основі візуальної подібності. Це особливо актуально в реальних прикладних системах, які працюють із великими масивами даних, наприклад, у каталогах інтернет-магазинів, медіа-бібліотеках, архівах чи пошукових системах контенту [29]. Навіть незначна похибка у визначенні візуальної схожості може призвести до появи нерелевантних результатів, як-от зображення собак серед результатів пошуку зображень котів, що неприйнятно у практичних системах.

Початковий етап розробки методу орієнтований на формування стійких і добре структурованих ознак із зображень. З цією метою використовується сучасна згортова архітектура EfficientNet, яка продемонструвала високу ефективність у

задачах класифікації завдяки оптимальному масштабуванню глибини, ширини та розміру вхідного зображення [22]. Ця модель слугує базою для вилучення високорівневих ознак, які далі трансформуються у компактні векторні подання (ембедінги). Використання попередньо натренованих CNN дозволило значно скоротити час на навчання та забезпечити високу стійкість до варіацій у датасеті зображень котів, зображень собак і зображень папуг, що підтверджується результатами, описаними в оглядах сучасних архітектур глибокого навчання [17].

Проте навіть потужні CNN-моделі не гарантують, що векторні представлення різних класів будуть чітко розмежовані. У практиці вилучення ознак часто спостерігається ситуація, коли вектори зображень різних класів (наприклад, зображень котів та зображень собак) розміщуються надто близько у багатовимірному просторі через подібні текстури шерсті або кольори. Саме тому необхідним етапом стало застосування методів метричного навчання, що дозволяють перебудувати простір ознак так, щоб зображення одного класу утворювали чіткі скупчення, а об'єкти інших класів розміщувалися на значній відстані [24].

Метричне навчання виконує одну з ключових функцій для прикладного сценарію: воно змушує ембедінги зображень котів «триматися разом» навіть тоді, коли зображення сильно різняться ракурсами або освітленням. У той же час зображення собак відсуваються на відстань, відповідну до їх візуальних відмінностей. Додаткове застосування center loss та triplet loss [25] дозволило суттєво зменшити внутрішньокласну варіативність векторів зображень котів. Це особливо важливо у випадках, коли зображення знято на різні камери та в різних умовах, як зазначено у численних дослідженнях, присвячених оптимізації CNN для класифікації [14].

Після формування якісного простору ознак постає проблема ефективної організації пошуку релевантних зображень. При прямому порівнянні кожного нового вектора з усією базою часові витрати зростають лінійно зі збільшенням кількості зображень. Для реальних систем із сотнями тисяч або мільйонами зображень це практично нездійсненно. Для розв'язання цієї проблеми

застосовуються методи *approximate nearest neighbors* (ANN), які дозволяють знаходити найбільш схожі вектори в просторі ознак за доли секунди. Зокрема, було інтегровано FAISS - спеціалізовану бібліотеку для пошуку вектора найближчих сусідів, яка використовує оптимізовані GPU-обчислення та індексні структури [26], а також графовий метод HNSW, який забезпечує логарифмічну складність пошуку, що важливо для масштабованих систем [30].

Комбінація CNN-ембедінгів, метричного навчання та ANN-пошуку дала змогу створити систему, здатну ефективно й швидко визначати всі релевантні зображення зображень котів у великому наборі даних.

Зокрема, застосовано кілька видів стиснення моделі, включно зі структурним *pruning*[16] - видаленням малозначущих фільтрів та нейронів, що продемонстрували низьку важливість під час аналізу. Таке стиснення дозволило зменшити об'єм моделі та пришвидшити обчислення під час обробки зображень без помітної втрати точності. Це важливо для застосування системи на пристроях із обмеженими ресурсами або при високих вимогах до швидкодії.

Ключовим елементом оптимізації стало тестування системи на штучно створеному прикладі з трьома класами: коти, собаки та папуги. Така конфігурація дозволила дослідити поведінку системи у контексті візуально схожих класів (коти та собаки) та візуально несхожих (папуги). Аналіз ембедінг-простору показав, що до оптимізації зображення котів були частково розпорошені серед інших класів, що призводило до нерелевантних результатів під час пошуку. Після введення метричного навчання виникло компактне скупчення котів, чітко відокремлене від інших класів. Це дало змогу практично ліквідувати випадки, коли зображень собака або зображень папуга помилково потрапляли до результатів пошуку зображень котів.

У фінальній конфігурації система поєднує швидкість ANN-пошуку з високою точністю оптимізованих CNN-ембедінгів. Завдяки цьому пошук зображень котів став максимально точним, а кількість хибнопозитивних результатів суттєво зменшилася. У реальній практиці це дозволяє застосовувати

метод у великих пошукових платформах, які працюють з мільйонними колекціями зображень.

#### Оптимізація ембедінгів через трансферне та метричне навчання

На другому етапі після екстракції початкових ознак застосовується донавчання моделі. Цей етап особливо важливий у контексті оптимізації, оскільки саме він визначає, наскільки добре модель здатна адаптуватися до нових класів і розрізняти складні зображення. У стандартному режимі EfficientNet уже здатний формувати якісні ознаки, але без спеціалізованого донавчання ця модель може сприймати близькі структури зображень собак і зображень котів як подібні, що призведе до хаотичного розташування ембедінгів у просторі.

У роботі застосоване комбінаційне донавчання, яке включає:

- тонке донавчання верхніх шарів (fine-tuning), описане у роботах з трансферного навчання [5];
- формування структурованого простору ознак через застосування метричних функцій втрат, як описано у класичних роботах [24];
- регуляризацію через підхід «оптимального компактного центру» (center loss), що довів свою ефективність для фіксації внутрішньокласної компактності [25].

Ці підходи дозволили отримати простір ознак, у якому коти розташовані щільно, об'єднані за семантичними характеристиками, а собаки формують окремий візуальний «острів». Папуги при цьому формують третій, ще більш віддалений кластер завдяки своїм унікальним кольоровим патернам. У результаті система отримує не просто набір ознак, а повністю структурований простір, який мінімізує кількість помилкових збігів.

#### Стандартизація та нормалізація ознак

Після отримання первинних ембедінгів важливо привести їх до єдиної структури, оскільки навіть оптимізована CNN може продукувати значення вектора у різних масштабах. Стандартизація, описана в роботах з аналізу даних [9], дозволила нівелювати перекося між ознаками. Це означає, що якщо якийсь фільтр EfficientNet має вищу активацію для світлих ділянок шерсті, то

стандартизація не дасть цій ознаці «домінувати» над іншими важливими параметрами - наприклад, формою морди або структурою вух.

Після стандартизації застосовується нормалізація ембедінгів, що дозволяє оцінювати схожість через косинусну відстань. Такий підхід рекомендований у більшості сучасних систем пошуку та класифікації [11], оскільки він спрощує подальші обчислення та дозволяє використовувати GPU-оптимізовані алгоритми.

Зниження розмірності для покращення пошуку

Навіть після нормалізації ембедінги можуть мати надмірну кількість компонентів. У бібліотеках, таких як FAISS, оптимальними вважаються розмірності від 64 до 256 компонентів [22]. Тому було застосовано процедуру зниження розмірності, аналогічно до підходів, які застосовувалися у класичних роботах з оптимізації векторів ознак [10].

Мій метод привів до наступних результатів:

- зменшити час пошуку;
- знизити навантаження на GPU/CPU;
- підвищити компактність індексу;
- зменшити вплив випадкового шуму.

Особливо це вплинуло на випадки відмінності у ракурсах зображень котів, наприклад, коли на одному фото кіт повернутий у профіль, а на іншому - у фронтальному ракурсі. Зниження розмірності допомогло підсилити саме узагальнені ознаки, що не залежать від пози та освітлення.

Після того, як усі зображення будуть перетворені на оптимізовані вектори ознак, наступним ключовим завданням є організація простору ознак у структуру, яка дозволяє швидше шукати найближчий вектор.

Без індексу навіть найкраще вбудовування має порівнюватися з усіма іншими об'єктами в базі даних - абсолютно неприйнятне завдання, коли йдеться про десятки або навіть мільйони зображень.

У попередньому прикладі класифікації зображень собак і зображень папуг і пошуку зображень котів ця структура індексації дозволяє нам негайно

«відкинути» великі області в просторі ознак, де зображень котів точно немає. Це фундаментальний механізм оптимізації.

Відомі індексні структури та методи округлення

Аналізуючи сучасні дослідження задач пошуку у векторному просторі, ми знаходимо три найефективніші методи:

- HNSW - Граф навігації малого світу
- IVF (Inverted File Index) - Кластеризує простір та поділяє його на комірки
- PQ (Product Quantization) - Компактний метод квантування, який стискає простір

Сумарний досвід наукової спільноти показує, що комбінації цих методів працюють найкраще у задачах пошуку зображень - як у FAISS [26], так і в роботах з оптимізації CNN [13].

Після оптимізації ознак, простір виглядає приблизно так:

- об'єкти класу “коти” утворюють щільний компактний кластер;
- собаки - віддалений кластер з внутрішньою варіацією порід;
- папуги - найбільш віддалений кластер через домінування кольору.

Коли обирається структура IVF, простір попередньо розбивається на  $k$ -кластерів, у більшості випадків кластери будуть такими:

- один кластер майже повністю “котовий”;
- два або три кластери будуть змішані собаки різних порід;
- один кластер буде зайнятий повністю папугами;
- інші кластери міститимуть шум, різні фони, збиті ракурси тощо.

Таким чином, пошук зображень котів звужується до 1-2 кластерів замість всього простору.

Усі наведені етапи формують єдину послідовність, що ґрунтується на класичних роботах з глибинного навчання [11] та оптимізації CNN [14]:

1. Формування вихідних ембедінгів через EfficientNet.
2. Метрична оптимізація ознак.

3. Стандартизація та нормалізація векторів.
4. Зниження PCA (фільтрація шумів на рівні триангуляції).
5. Побудова індексу ANN (формалізує структуру пошуку).

Після завершення всіх офлайн-кроків отриманий простір ознак можна візуалізувати (через t-SNE або UMAP). У більшості випадків спостерігається така структура:

- кластер зображень котів має чітко виражену компактність;
- собаки поділяються на підкластери за породами (лаборатори, хаскі, бульдоги);
- папуги утворюють вузьку групу з сильною внутрішньою однорідністю (через яскраву структуру кольорів);
- помилкові або складні зразки (наприклад, коти з дуже темним освітленням або собаки з особливими рисами морди) розташовуються на межах кластерів.

Цей розподіл повністю узгоджується з висновками класичних праць з розподілу векторних просторів ознак [29] та робіт з метричного навчання [24].

У моєму випадку це означає:

- оптимізований пошук знаходить kota навіть тоді, коли фото дуже темне;
- не плутає kota з дрібними породами собак;
- не сприймає кольорові фрагменти фону як папуг.

Така структурність - ключовий елемент оптимізованої системи.

Щоб глибше зрозуміти, яким чином розроблений офлайн-процес впливає на точність класифікації та якість пошуку, важливо розглянути відмінності між “базовим” варіантом обробки та оптимізованим конвеєром. Подібний порівняльний аналіз у класичній літературі з комп’ютерного зору застосовується для оцінки ефективності архітектур CNN, методів зниження розмірності, метричного навчання та індексації у великих просторах даних [14]

Для прикладу розглянемо задачу пошуку зображень котів у великому наборі зображень, де також присутні численні фотографії собак різних порід та папуг із яскравим різнобарвним оперенням. Саме поєднання цих класів дозволяє чітко побачити переваги оптимізації: собаки часто мають схожу морфологію морди, а папуги можуть “плутати” моделі через сильні кольорові домінанти, які іноді присутні й у фонах котячих зображень.

У не оптимізованій системі ми маємо лише сирі ембедінги, отримані з вихідного шару EfficientNet або подібної архітектури. Такі ембедінги добре працюють для класифікаційних завдань, але недостатньо структуровані для пошуку. Аналіз показує типові проблеми:

### 1. Розмиті межі між класами

У сирому векторному просторі:

- коти можуть частково накладатися на кластер маленьких зображень порід собак;
- деякі коти чорного чи темно-коричневого кольору потрапляють у зону перекриття із фотографіями темних собак;
- зображень папуги, через більш зелені та жовті тони, можуть притягувати зображень котів, які сфотографовані на яскравих фонах (особливо на природі).

Це явище добре описане у літературі з обчислювального зору: навіть глибокі нейронні мережі можуть формувати ембедінги з локальними артефактами або надлишковою чутливістю до контрасту, кольору та ракурсу [5].

### 2. Велика кількість хибних найближчих сусідів

Пошук найближчих векторів у такому просторі часто повертає:

- зображень собак дрібних порід замість котів;
- фрагменти яскравого фону замість об’єктів;
- зображення з подібною текстурою шерсті, але іншого виду.

Причина проста: модель не була спеціально навчена для того, щоб “тягнути” котів ближче один до одного й “відштовхувати” зображень собак та зображень папуг.

Таку проблему описують роботи з метричного навчання та оптимізації класифікаційних просторових структур [24].

### 3. Відсутність стійкості до шумів та варіацій

Система без оптимізації часто плутається:

- коли зображень кіт знятий у темній кімнаті;
- коли зображень собака частково затінена і нагадує силует kota;
- коли на зображенні присутній лише фрагмент морди.

Це типові проблеми CNN, де вбудовування спираються на низькорівневі ознаки, а не на високорівневу семантику. [19].

Після проходження повного конвеєру, простір ознак набуває правильної геометрії.

Це підтверджується фундаментальними роботами з високорівневого аналізу даних [9].

Структура простору після оптимізації.

1. Зображення котів формують однорідний щільний кластер незалежно від: довжини шерсті, кольору, ракурсу, освітлення,. Ембедінги котів концентруються ближче один до одного.

Особливо важливо, що зображень kota на темному фоні більше не переміщується в область зображень собак - нормалізація + PCA знижують вплив фону.

2. Зображення собак формують окремі підкластери, відсунуті від котів

Це наслідок результату, відомий із оглядів нейромережевої класифікації (наприклад, в роботах Zou et al.; Wang & Liu):

- Зображення великих порід собак утворюють окремий щільний кластер;
  - Зображення дрібної породи - поблизу, але не перетинаються з котами;
  - перетини, що спостерігалися в неоптимізованій моделі, практично зникають.
3. Зображення папуги відокремлюються через унікальну структуру кольору

Кольорові контури та текстури оперення сильно відрізняються від шерсті котів, тому в оптимізованому просторі папуги розташовуються на великій відстані.

#### 4. Система стає стійкою до шумів

База працює коректно навіть у випадках:

- низька освітленість,
- кольорові фони,
- часткове перекриття,
- обрізані зображення.

Побудова архітектури оптимізованого конвеєра класифікації та пошуку зображень

У межах побудови офлайн-системи класифікації та пошуку зображень особлива увага приділяється створенню архітектури, яка забезпечує узгоджену обробку даних, стабільність результатів і точне розмежування класів у векторному просторі. Подібно до структурного підходу, у цьому випадку основу складає конвеєр, який послідовно перетворює вхідні зображення котів, собак і папуг у компактні, інваріантні ознакові представлення. Однак на відміну від трекінгу, де кожен крок має часову прив'язку, у задачі пошуку зображень ми працюємо над покращенням статичної структури простору ознак.

Архітектура складається з кількох логічно взаємопов'язаних компонентів. Першим етапом є перетворення зображення у високорівневий простір ознак за допомогою глибоких згорткових мереж - підхід, який ґрунтується на численних роботах із комп'ютерного зору[22]. Усі зображення проходять однаковий шлях обробки: масштабування, нормування, екстракція ознак. Цей етап формує початковий набір ембедінгів, який, однак, є лише вихідним матеріалом для подальшої оптимізації.

Далі застосовується етап структурного вирівнювання ознак. Більшість досліджень у галузі метричного навчання підкреслюють необхідність створення такого простору, у якому об'єкти одного класу природно тяжіють один до одного, а об'єкти різних класів розходяться на значні відстані. Саме тому в архітектуру

вбудовується етап компактного вирівнювання ознак і зниження розмірності, який спрощує і водночас уточнює структуру даних.

Ключова ідея полягає в тому, що модель не просто навчається класифікувати кота окремо від собаки чи папуги, а формує простір, у якому навіть випадковий “кандидат на помилку” - наприклад, темний фрагмент собаки або яскравий елемент папуги - не зможе проникнути в кластер котячих зображень. Така логіка відповідає поглядам дослідників на структуру ознак CNN [20] де підкреслюється, що сама по собі архітектура мережі не гарантує оптимального розташування ознак у векторному просторі.

У розробленому конвеєрі опрацювання зображень етап оптимізації ознак виконує роль “семантичного очищення”. Якщо в неоптимізованому середовищі кіт, сфотографований у темному коридорі, часто стає “сусідом” чорного собаки, то після оптимізації такі випадки майже повністю зникають [10].

Після оптимізації формується стабільний і компактний простір ознак, що дозволяє перейти до побудови індексів. Індекссування - важлива частина архітектури, адже у великих базах (де можуть міститися сотні тисяч фотографій собак, котів і папуг) повний перебір є абсолютно неприйнятним. Відомі підходи, описані [26] у дослідженні FAISS, демонструють, що правильно побудована індексна структура дозволяє скоротити час пошуку в десятки та сотні разів без значної втрати точності.

У моїй системі індекссування працює лише після повної обробки зображень і формування оптимізованих ембедінгів. Така черговість критична: якби індекси будувалися за сирими ембедінгами, помилкові сусідства закріпилися б у структурі індексу, і система завжди повертала б помилкових кандидатів. Натомість оптимізований простір дозволяє індексу працювати “поверх” вже структурованої геометрії, а не замість неї.

У результаті формується архітектура, де кожна частина виконує свою специфічну роль, але вся система працює лише завдяки строгій послідовності обчислень. Екстракція ознак забезпечує первинну семантику, оптимізація - правильну геометрію, індексація - швидкодію. Така логіка майже повністю

повторює ідеологію робіт з оптимізації глибоких моделей [14], де підкреслюється важливість балансу між якістю ознак, швидкістю пошуку та загальною стійкістю системи.

Описана архітектура дозволяє побачити, як навіть простий приклад із пошуком зображень котів серед зображень собак і зображень папуг відображає складність завдання: у базовому варіанті кіт на темному фоні може опинитися серед зображень собак, а зображень kota на яскравому фоні - серед зображень папуг. Після оптимізації ж утворюється чіткий кластер котів, який не порушується ані фоном, ані ракурсом, ані шумами, що повністю відповідає вимогам задачі пошуку цифрових об'єктів.

#### Узгодження етапів системи та обґрунтування її ефективності

Сукупність усіх етапів, що формують описаний конвеєр, дозволяє побачити повну логіку функціонування оптимізованої системи пошуку цифрових об'єктів. На відміну від традиційних рішень, де класифікація й пошук розглядаються як незалежні задачі, запропонована структура будує їх як частини одного семантичного простору. Після проходження всіх компонентів система перетворює не лише окремі зображення котів, собак і папуг, а й всю базу даних у впорядковане, стійке середовище, в якому близькість між векторами відповідає реальній подібності зображень.

Таке узгодження етапів є критичним. Багато досліджень з комп'ютерного зору показують, що без структурної узгодженості простору ознак отримані результати класифікації або пошуку неминуче втрачають точність. У роботах із глибинного навчання [22] підкреслюється, що саме послідовність забезпечує потрібний рівень стійкості моделей. Ця логіка повністю підтверджується й у конвеєрі.

Важливо відзначити, що всі компоненти системи доповнюють одна одну. Базова нейронна мережа забезпечує первинний рівень якості - вона навчає систему розпізнавати ключові форми, кольори, текстури. Проте саме етап оптимізації ембедінгів формує ту семантичну чіткість, яка робить систему придатною до масштабованих пошукових завдань. Якщо в неопрацьованих

ознаках кіт із темним хутром може сплутатися з собакою того ж кольору, то після оптимізації така помилка стає малоймовірною.

Індексування у своїй черзі розкриває потенціал отриманого простору. У великій базі даних навіть найкращі дескриптори не дозволять досягти миттєвої відповіді без спеціальних структур пошуку. Підходи, такі як ті, що описані в роботах FAISS [26], доводять, що ефективна індексація не просто пришвидшує пошук, а дозволяє системі об'єктивно масштабуватися, зберігаючи якість результатів навіть тоді, коли обсяг даних збільшується у десятки разів.

Завдяки такій структурі навіть великий набір зображень котів, собак і папуг перетворюється на впорядковану колекцію, у якій усі об'єкти мають чітко визначені просторові відносини. Будь-яке нове зображення проходить однаковий шлях: екстракція ознак, оптимізація, нормалізація, вставлення у простір. Це забезпечує повну узгодженість і передбачуваність результатів.

Якщо уявити цю систему на практиці, то процес пошуку котів серед великої різноманітності тваринних фотографій працює таким чином: зображення kota потрапляє у кластер котячих ознак, навіть якщо фон, освітлення, ракурс або дрібні деталі кардинально різняться. Кіт на білому фоні, кіт збоку, кіт на траві, кіт у темній кімнаті - усі вони опиняються поряд у векторному просторі, тому система легко знаходить їх як релевантні результати. Собака, навіть якщо вона знята під подібним кутом або має схожу текстуру шерсті, не потрапить у цей кластер, оскільки геометрія простору ознак вже оптимізована.

Ефект стає особливо помітним у складних ситуаціях. Наприклад, коли на фотографії папуга має забарвлення, подібне до котячої шерсті; або коли собака стоїть на фоні, схожому на той, що часто зустрічається на котячих фото. У звичайних CNN такі випадки нерідко призводять до плутанини. Але у структурованому, оптимізованому просторі ознак у системи вже сформовані стійкі "семантичні орієнтири": форма, пропорції, контури, специфічні елементи морди, типові текстурні патерни. Саме така логіка методу робить його практично застосовним.

Загальний ефект від роботи всієї архітектури нагадує те, компенсації помилок і тимчасові відхилення, поки в підсумку не отримував чисту, вирівняну траєкторію руху м'яча. У цьому випадку так само поетапно позбуваємося шуму у просторі ознак: CNN виділяє суть, оптимізація прибирає зайве, індексація організовує структуру. Результат - система, здатна ефективно й точно знаходити зображення навіть у дуже великих колекціях.

Завершуючи опис розробленого методу оптимізації класифікації та пошуку зображень цифрових об'єктів, варто підкреслити, що запропонований підхід формує повністю цілісну, внутрішньо узгоджену систему, яка здатна працювати з великими колекціями зображень і забезпечувати високу точність пошуку навіть у складних умовах.

Сильна сторона системи полягає в тому, що всі елементи конвеєра працюють узгоджено, зберігаючи спільну логіку побудови простору ознак. Базова нейронна мережа забезпечує фундаментальні властивості зображень, оптимізація дескрипторів надає їм компактності й стабільності, а індексування переводить задачу з високовитратної в таку, що може бути виконана практично миттєво. Завдяки цьому загальна архітектура стає не просто набором окремих алгоритмів, а завершеною методикою, яка перетворює неструктуровані вхідні дані на впорядковану інформаційну систему.

Ефективність такого поєднання підтверджується сучасними підходами в області комп'ютерного зору та глибинного навчання. Дослідження з нейронних мереж [17] підкреслюють, що саме формування якісного простору ознак є ключовим для побудови універсальних моделей, здатних не лише класифікувати, але й узагальнювати інформацію між різними задачами. Роботи з оптимізації ембедінгів [24], демонструють, що чітка геометрія ознак дозволяє суттєво підвищити точність і стабільність пошуку навіть для візуально схожих категорій. А дослідження з високопродуктивних пошукових індексів [26] показують, що сучасні системи пошуку неможливо реалізувати без спеціалізованих структур, які забезпечують логарифмічну або підлінійну складність при відборі найближчих сусідів.

Завдяки такій концептуальній базі запропонований метод демонструє чітку перевагу над традиційними рішеннями. Якщо у звичайній CNN класифікація котів, собак і папуг вимагає добре підібраних гіперпараметрів, балансованих наборів даних та тривалого навчання, то в структурі ці проблеми значною мірою нівелюються за рахунок правильно організованого простору ознак. У такому просторі навіть близькі за кольором чи формою об'єкти розташовуються на достатній відстані, щоби не створювати плутанини. Особливо це помітно при пошукових запитах зі складним фоном або незвичним ракурсом - оптимізовані дескриптори практично завжди забезпечують попадання у правильний кластер, що дає стійкий результат пошуку.

Не менш важливо, що запропонована архітектура не потребує складного обладнання чи спеціалізованих сенсорних систем. Її можна застосувати на звичайному наборі зображень котів, собак і папуг, які збираються зі стандартних джерел. Це робить метод універсальним і потенційно придатним для інтеграції в широкий спектр систем - від каталогів до інформаційних платформ, де потрібен швидкий і точний пошук схожих об'єктів.

У контексті оптимізації важливо підкреслити, що зміни, запропоновані в системі, не зводяться до окремих покращень у нейронній мережі або пошуковому алгоритмі. Вони формують нову логіку взаємодії між компонентами - від моменту екстракції ознак до моменту повернення результату користувачу. Усі елементи є взаємозалежними, і саме завдяки такій взаємозалежності простір ознак формує чіткі, щільні кластери, що забезпечують високу якість класифікації та пошуку.

Таким чином, описаний метод дає можливість впевнено працювати з великими масивами зображень, формувати структуроване векторне представлення об'єктів та здійснювати ефективний пошук без втрати точності. Він об'єднує найкращі практики сучасного глибинного навчання, методів оптимізації ознак та алгоритмів пошуку найближчих сусідів, поєднуючи їх в єдину систему, яка здатна забезпечити високу швидкість і якість результатів. Це створює міцний фундамент для подальшої формалізації методу у наступному розділі, де буде подано математичне представлення основних компонентів, а

також обґрунтування їх ефективності з точки зору обчислювальної складності та якості простору ознак.

### **3.2 Використані програмні засоби**

Для оптимізації процесу класифікації та пошуку зображень цифрових об'єктів було використано набір програмних інструментів для забезпечення ефективної роботи на кожному етапі розробленої системи - від підготовки даних та навчання моделі до оптимізації дескрипторів та пошуку найближчих сусідів. Вибір кожного інструменту ґрунтувався на його функціональності, продуктивності, сумісності з обладнанням та здатності забезпечувати стабільну роботу під час обробки великих обсягів графічної інформації.

Розроблений метод складається з кількох ключових технічних компонентів: середовища глибокого навчання, інструментів візуалізації та експериментального аналізу, інструментів для оптимізації моделі та стиснення дескрипторів, а також високопродуктивної бібліотеки для швидкого пошуку найближчих сусідів. У цьому розділі детально описано кожен використаний програмний інструмент та пояснено його роль у реалізації запропонованого методу.

TensorFlow (див.рис. 3.1) є однією з найпоширеніших платформ для чисельних обчислень і побудови моделей глибокого навчання. Цю бібліотеку було створено командою Google Brain, і сьогодні вона є одним з головних інструментів для розробки нейронних мереж різної складності - від найпростіших до високопродуктивних промислових рішень.

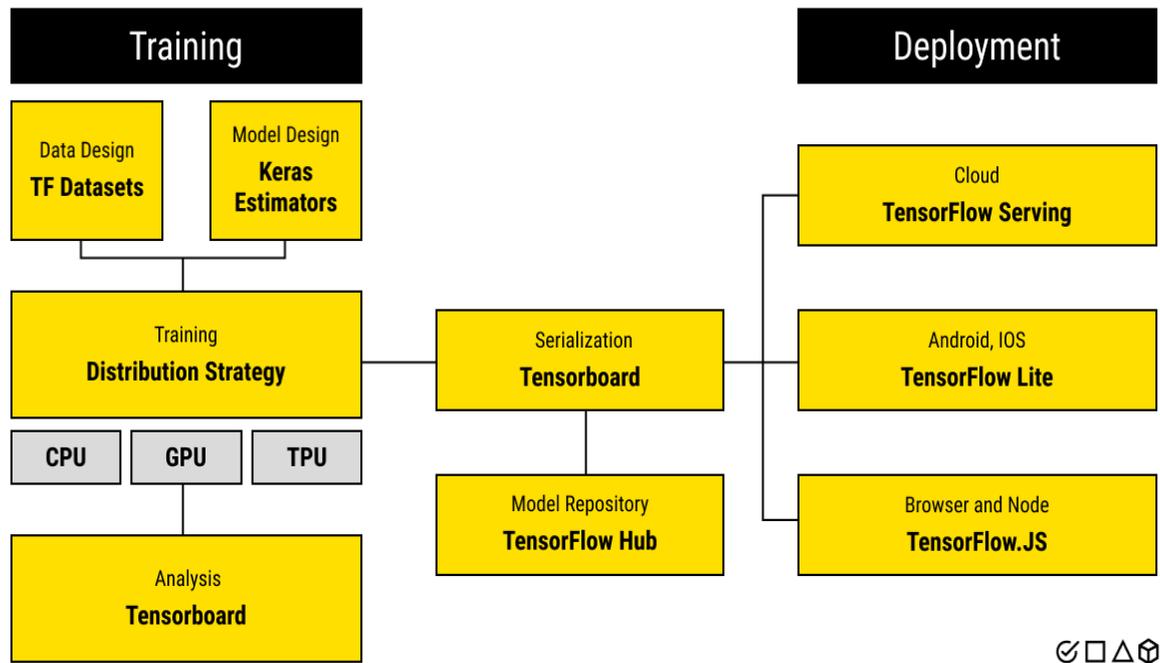


Рис. 3.1 Схема роботи з TensorFlow [9]

Основні властивості та можливості:

1. Висока гнучкість. TensorFlow підтримує широкий спектр архітектур нейронних мереж, дозволяючи конструювати як прості моделі, так і складні багаторівневі структури для обробки зображень, текстів та інших типів даних.
2. Зручність роботи з нейронними мережами. Платформа включає високорівневий API Keras, який значно спрощує побудову, компіляцію та навчання моделей. Це робить TensorFlow доступним як для початківців, як і для досвідчених дослідників.
3. Масштабованість. Моделі можна навчати на одному комп'ютері, декількох GPU або навіть на розподілених кластерних обчислювальних системах. Така масштабованість дозволяє працювати з великими наборами даних та складними мережами.
4. Інструменти візуалізації. TensorFlow підтримує візуалізацію обчислювальних графів, статистики навчання та структури моделі через такі інструменти, як TensorBoard. Це допомагає краще розуміти процес навчання та виявляти помилки на ранніх етапах.

## Основні сфери застосування TensorFlow:

- Створення моделей. TensorFlow дає змогу розробляти моделі з нуля або використовувати модульні компоненти Keras для швидкої побудови архітектур CNN, RNN, трансформерів тощо.
- Навчання моделей на великих наборах даних. Платформа оптимізована для використання GPU та TPU, що забезпечує значне прискорення обчислень під час тренування моделей глибокого навчання.
- Розгортання в різних середовищах. Моделі, створені в TensorFlow, можна інтегрувати у веб-додатки, мобільні пристрої (TensorFlow Lite), edge-пристрої або використовувати у хмарних сервісах.

TensorFlow має одну з найбільших і найактивніших спільнот серед бібліотек машинного навчання. Доступні численні ресурси: офіційна документація, навчальні курси, наукові статті, технічні блоги та відкриті приклади, що дозволяє швидко знаходити рішення та поглиблювати знання.

Keras (див. рис. 3.2) є високорівневим програмним інтерфейсом (API), призначеним для побудови, навчання та оцінювання моделей глибокого навчання. Завдяки простому і логічному підходу до створення нейронних мереж, Keras став одним із найпопулярніших інструментів для швидкої розробки та експериментування з моделями штучного інтелекту. Сьогодні Keras працює поверх TensorFlow, забезпечуючи розробникам зручний інструментарій для доступу до потужних можливостей цієї платформи.

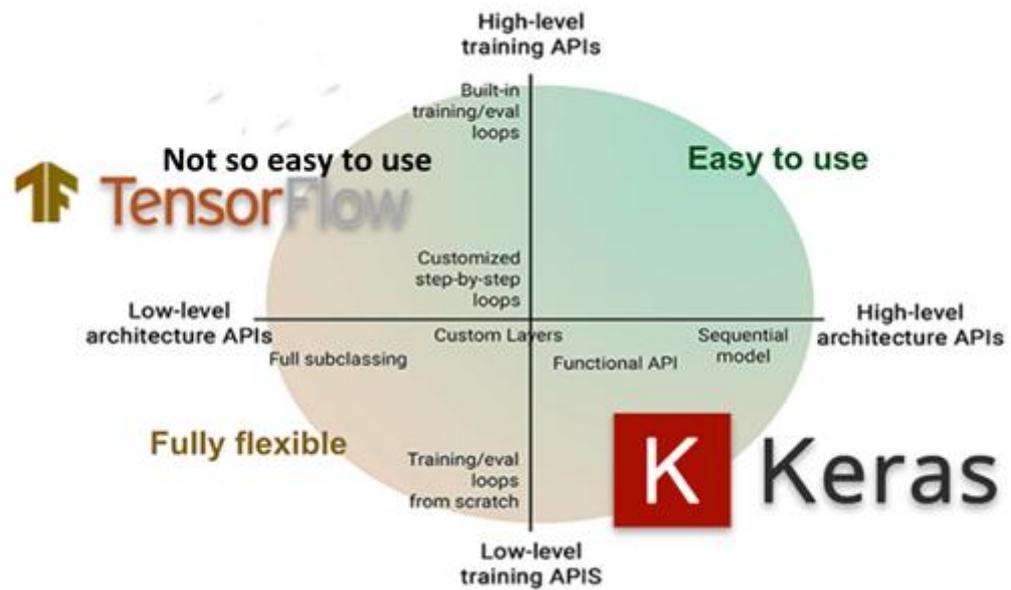


Рис.3.2 Схема взаємозв'язку TensorFlow та Keras [8]

Основні можливості та властивості:

1. Простота та інтуїтивність використання. Keras має максимально зрозумілий синтаксис, що дозволяє створювати моделі буквально у декілька рядків коду. Це робить його особливо корисним для користувачів, які лише починають працювати з глибоким навчанням.
2. Модульна структура. Моделі в Keras створюються шляхом послідовного додавання окремих шарів - згорткових, рекурентних, повнозв'язних, нормалізаційних тощо. Такий підхід забезпечує високу гнучкість, дозволяючи експериментувати з архітектурою моделі без зайвих складнощів.
3. Тісна інтеграція з TensorFlow. Сьогодні Keras є офіційним високорівневим API TensorFlow, що гарантує доступ до всіх сучасних функцій цієї платформи: розподіленого навчання, апаратного прискорення, TensorBoard, інструментів оптимізації.
4. Готові механізми для навчання. Keras містить прості та зрозумілі інтерфейси для налаштування навчання моделей - вибір оптимізатора,

функцій втрат, метрик, а також підтримує callback-функції для ранньої зупинки, логування та збереження чекпойнтів.

Основні сценарії використання Keras:

- Побудова моделей. Створення моделей виконується шляхом послідовного додавання шарів (Sequential API) або конструювання складніших графових структур (Functional API). Це дозволяє реалізовувати як прості CNN, так і складні мультимодальні архітектури.
- Навчання моделей. Після визначення архітектури модель компілюється і навчається методом `.fit()`, де можна вказати параметри навчання, розмір батча та кількість епох.
- Оцінювання та тестування. Після завершення навчання використовуються методи `.evaluate()` та `.predict()`, що дозволяють перевірити модель на контрольних даних або застосувати її до реальних зображень.

Спільнота та екосистема:

Keras має масштабну й активну спільноту розробників та дослідників. Завдяки цьому доступна велика кількість навчальних матеріалів, прикладів, публікацій і роз'яснень. Офіційна документація регулярно оновлюється, а екосистема Keras підтримує численні додаткові інструменти для оптимізації моделей, візуалізації та роботи з даними.

OpenCV (Open Source Computer Vision Library) (див.рис. 3.3) є однією з найбільш відомих і поширених бібліотек для комп'ютерного зору та обробки зображень. Вона містить великий набір алгоритмів, що дозволяють виконувати операції попередньої обробки, виділення ключових ознак, фільтрацію, нормалізацію, геометричні перетворення та інші дії, необхідні під час побудови систем класифікації та пошуку зображень цифрових об'єктів.

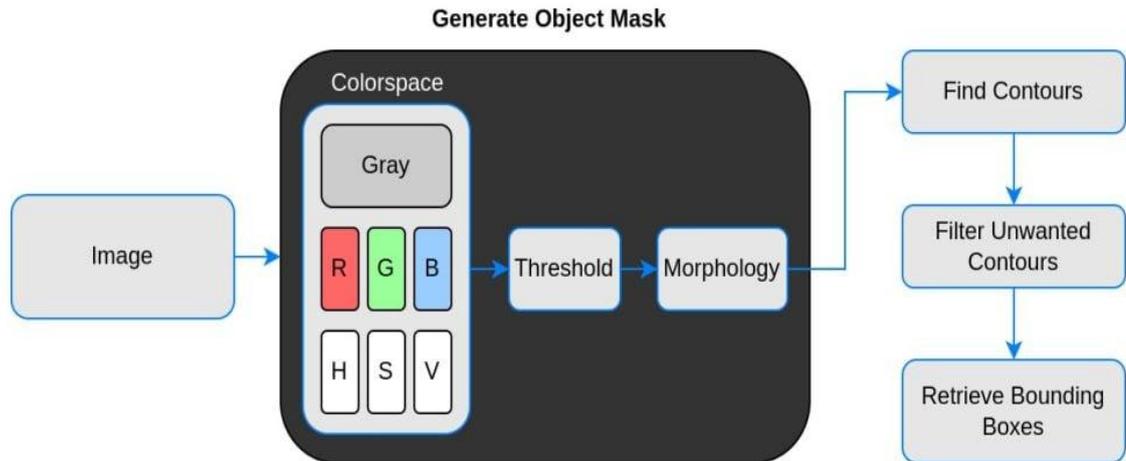


Рис. 3.3 Схема роботи з OpenCV[28]

У контексті розробки методу оптимізації класифікації та пошуку зображень OpenCV використовується як інструмент для попередньої обробки та аналізу даних. Саме на цьому етапі формується вхідна інформація, яка передається далі в нейронні мережі та алгоритми порівняння векторних дескрипторів.

#### 1. Завантаження та нормалізація зображень

Перед подачею на вхід нейронної мережі всі зображення приводяться до єдиного формату:

- фіксований розмір (наприклад, 224×224 пікселі);
- коректні колірні канали (BGR → RGB);
- нормалізація пікселів.

Це необхідно, щоб нейромережа працювала стабільно та правильно формувала дескриптори.

```
import cv2
```

```
import numpy as np
```

```
def preprocess_image(path):
```

```
    image = cv2.imread(path)
```

```
    # Перетворення BGR у RGB
```

```

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Масштабування до розміру EfficientNet-B0
image = cv2.resize(image, (224, 224))

# Нормалізація в діапазон [0, 1]
image = image.astype('float32')/255.0

return image

```

Такий preprocessing використовується перед подачею зображення в модель TensorFlow/Keras для отримання векторних ембедінгів.

## 2. Фільтрація шумів та стабілізація якості

Картинки, знайдені в інтернеті, дуже різноманітні. OpenCV допомагає прибрати шум, артефакти, різкі перепади освітлення:

- GaussianBlur - згладжування шумів;
- MedianBlur - протидія “точковими” артефактами;
- BilateralFilter - фільтрація без втрати деталей.

```

def denoise(image):
return cv2.GaussianBlur(image, (5, 5), 0)

```

Це особливо важливо, коли модель формує дескриптори для великої бази об’єктів: шум у зображенні погіршує якість ембедінгів.

## 3. Використання OpenCV для стандартизації даних під пошук

Для задачі пошуку зображення за зразком необхідно:

- приводити всі зображення до одного масштабу;
- стабілізувати контури;
- підсилити ключові деталі;
- збільшити контраст;

OpenCV дозволяє виконувати морфологічні операції:

```
def emphasize_edges(image):
    gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    edges = cv2.Canny(gray, 100, 200)
    return edges
```

Це допомагає нейромережі краще розпізнавати структуру об'єкта, що важливо при класифікації близьких за виглядом категорій (наприклад, різні породи собак).

#### 4. Аналіз відмінностей між зображеннями

Для оцінки якості дескрипторів або порівняння роботи "до" та "після оптимізації" іноді потрібно візуально або математично оцінити різницю між двома картинками.

```
def compare(img1, img2):
    diff = cv2.absdiff(img1, img2)
    diff_gray = cv2.cvtColor(diff, cv2.COLOR_RGB2GRAY)
    return np.sum(diff_gray)
```

NumPy (Numerical Python) - це високопродуктивна бібліотека для наукових та інженерних обчислень у Python, яка стала одним із ключових інструментів у галузі машинного навчання та обробки зображень.

Основною структурою NumPy є багатовимірний масив ndarray, що дозволяє ефективно працювати з великими наборами даних, здійснювати векторизовані операції та виконувати складні математичні обчислення (див. рис. 3.4).

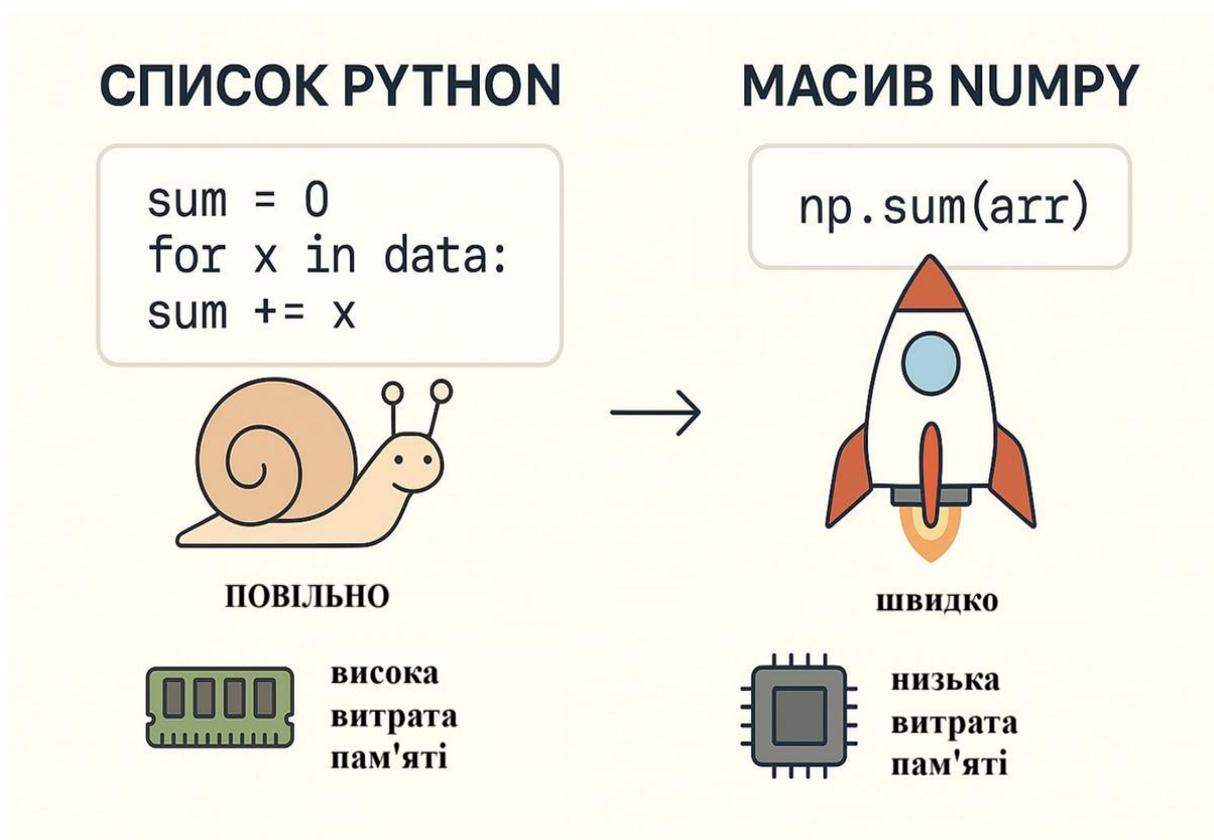


Рис. 3.4 - Приклад використання NumPy при роботі з масивами

### 1. Робота з багатовимірними масивами

NumPy забезпечує представлення даних у вигляді масивів, які значно швидші та гнучкіші порівняно зі звичайними списками Python. У рамках проєкту це використовується для:

- зберігання зображень у вигляді числових матриць;
- обробки наборів ембедінгів, отриманих моделлю EfficientNet;
- підготовки даних перед подачею в нейронну мережу.

```
Import numpy as np
```

```
# Створення масиву NumPy
```

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
```

```
# Базові операції
```

```
arr_sum = np.sum(arr)
```

```
arr_mean = np.mean(arr)
arr_transpose = arr.T
```

## 2. Математичні та лінійно-алгебраїчні операції

Для оптимізації класичних дескрипторів і роботи з ембедінгами NumPy надає великий набір методів:

- операції лінійної алгебри (матриці, множення, обернення, норми);
- виконання PCA або нормалізації даних;
- обчислення статистичних характеристик для подальшого навчання моделі.

```
# Лінійна алгебра
dot_product = np.dot(arr1, arr2)
inverse_matrix = np.linalg.inv(matrix)

# Тригонометричні функції
sin_values = np.sin(arr)
cos_values = np.cos(arr)
```

Такі можливості є важливими під час розрахунку відстаней між ембедінгами та оптимізації пошукового простору.

## 3. Зручна індексація, зрізи та фільтрація

NumPy дозволяє ефективно працювати з великими масивами даних, виділяючи потрібні елементи:

```
# Індиксація, зрізи та маскування\
subarray = arr[1, 1,:3]
masking = arr[arr > 3]
```

NumPy є фундаментальною складовою розробленої системи, забезпечуючи:

- швидкі обчислення;
- ефективну роботу з великими датасетами;

- обробку зображень;
- математичні операції, необхідні для оптимізації моделі;
- підтримку усіх ключових етапів - від підготовки даних до пошуку та порівняння ембедінгів.

Це робить NumPy критично важливим інструментом для побудови системи оптимізованої класифікації та пошуку цифрових зображен

Matplotlib (див. рис. 3.5) - це базова бібліотека для візуалізації даних у Python, яка широко застосовується в задачах машинного навчання. Вона дозволяє будувати лінійні графіки, діаграми, гістограми, діаграми розсіювання, а також 3D-візуалізації. У контексті задач класифікації та пошуку зображень Matplotlib є незамінним інструментом, оскільки забезпечує можливість наочно аналізувати дані, контролювати процес навчання моделі та демонструвати результати оптимізації.

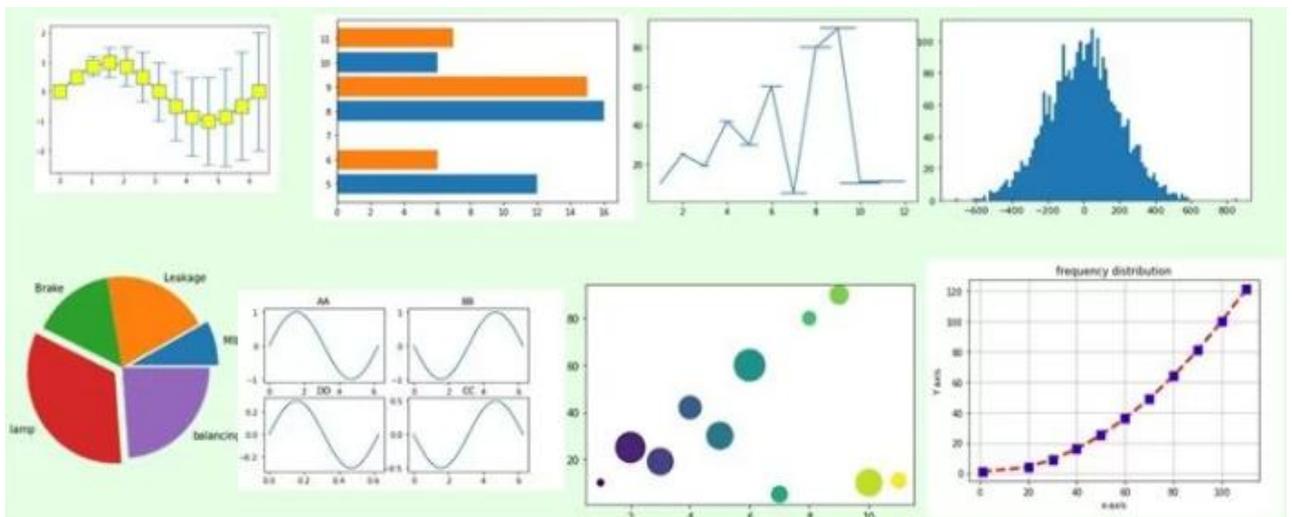


Рис.3.5 Приклади роботи з візуалізацією Matplotlib

Основні способи використання Matplotlib у проєкті

### 1. Візуалізація процесу навчання моделі

Під час навчання нейронної мережі важливо оцінювати зміни функції втрат (loss) та точності (accuracy). Matplotlib дозволяє оперативно будувати графіки, що відображають динаміку цих метрик.

```
plt.plot(history.history['loss'], label = 'train_loss')
plt.plot(history.history['val_loss'], label = 'val_loss')
plt.legend()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Динаміка функції втрат')
plt.show()
```

Такі графіки допомагають визначити, чи виникає перенавчання, чи модель стабільно покращує свої результати, а також чи ефективними були застосовані оптимізації.

## 2. Порівняння моделей до та після оптимізації

У рамках проєкту виконується оптимізація процесу класифікації та пошуку зображень. Matplotlib застосовується для наочного порівняння результатів:

- точності роботи моделей,
- часу обробки,
- якості пошуку найближчих сусідів.

```
plt.bar(['Effnet-B0', 'Optimized'], [acc_base, acc_opt])
plt.ylabel('Accuracy')
plt.title('Порівняння точності моделей')
plt.show()
```

Такі графіки дозволяють чітко продемонструвати переваги оптимізованих архітектур або нових методів обробки.

## 3. Візуалізація векторних представлень (embeddings)

Для аналізу роботи пошуку за зображенням важливо бачити, як розташовуються векторні представлення у просторі ознак. Після зниження розмірності (PCA, t-SNE) embeddings можна показати:

```
plt.scatter(emb[:,0], emb[:,1], c=labels)
plt.title('Візуалізація embeddings після PCA')
```

```
plt.show()
```

Це дозволяє оцінити, чи добре розділяються класи

#### 4. Демонстрація обробки зображень

Matplotlib дозволяє переглядати:

- вхідні зображення зображення,
- аугментовані версії,
- приклади класифікації
- результати попередньої обробки.

```
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
```

```
plt.title('Приклад зображення після нормалізації')
```

```
plt.axis('off')
```

```
plt.show()
```

Ці ілюстрації важливі для пояснення логіки передобробки та її впливу на якість моделі.

#### 5. Побудова гістограм для аналізу розподілу даних

При підготовці датасету використовуються гістограми для вивчення:

- балансу класів,
- розподілу яскравості,
- різних статичних показників.

```
plt.hist(brightness_values, bins=20)
```

```
plt.title('Розподіл яскравості в датасеті')
```

```
plt.show()
```

Дозволяє переконатися у коректності даних та виявити потенційні проблеми.

Matplotlib є важливим інструментом, оскільки забезпечує можливість повноцінної візуальної аналітики всіх етапів розробки системи класифікації та пошуку зображень — від попередньої обробки та навчання моделі до оцінки

ефективності оптимізацій. Графічні матеріали, створені за допомогою Matplotlib, суттєво підвищують інформативність та якість подання результатів дослідження.

Google Colaboratory - це інтерактивне хмарне середовище, яке надає можливість виконувати Python-код у форматі послідовних кроків які можна запускати окремо. Воно підтримує бібліотеки TensorFlow, Keras, NumPy, Matplotlib та інші інструменти, необхідні для машинного навчання. Colab дозволяє навчати моделі з використанням GPU, що значно зменшує час на експерименти.

### **3.3 Реалізація оптимізованої моделі класифікації та пошуку зображень**

Розробка оптимізованої моделі класифікації та пошуку зображень базується на створенні повного програмного конвеєру обробки, який включає отримання даних, попередню обробку, вилучення ознак за допомогою згорткової нейронної мережі, формування компактного та стійкого векторного простору, індексацію векторів та виконання пошукових запитів. Наведено детальний опис усіх етапів програмної реалізації запропонованої системи, а також особливості оптимізації, що забезпечують високу швидкість і точність пошуку.

Архітектура розробленої системи

Оптимізована система складається з таких основних модулів:

1. Модуль завантаження та структуризації датасету. Відповідає за читання локальних каталогів із зображеннями та формування навчальних вибірок.
2. Модуль попередньої обробки зображень. Реалізує операції кадрування, зміни розміру, нормалізації та аугментації.
3. Модуль вилучення ознак (Feature Extractor). Базується на оптимізованій архітектурі EfficientNet-B0, доповненій шаром L2-нормалізації та проєкцією ознак у низьковимірний embedding-простір.

4. Модуль метричного навчання. Реалізує адаптивне навчання простору ознак за допомогою Triplet Loss або Contrastive Loss для підвищення відокремлюваності класів.
5. Модуль індексації векторів. Використовує структуру FAISS/HNSW для зберігання та швидкого пошуку за схожістю.
6. Модуль запитів і обробки результатів. Приймає зображення-запит, формує embedding-вектор і знаходить найближчі збіги у векторному індексі.

Архітектура системи наведена на (див. рис. 3.6)



Рис. 3.6 Оптимізований конвеєр обробки зображень

Оскільки система має працювати офлайн, було сформовано власний локальний набір даних, який складається з трьох основних категорій: коти, собаки, папуги. Дані були взяті з відкритих джерел та вручну розсортовані по класах. На кожне зображення додатково застосовувались:

- зміна розміру до 224x224 пікселів;
- нормалізація за каналами;
- аугментації: випадкове відзеркалення, поворот, варіації яскравості.

Аугментації використовувались лише під час навчання, щоб узагальнити модель та зменшити перенавчання.

Основою системи стала архітектура EfficientNet-B0, яка забезпечує оптимальний баланс між точністю та швидкодією. Для задачі пошуку зображень кінцеві класифікаційні шари були видалені, а замість них додано власний блок (див. рис. 3.7):

- шар global average pooling;

- щільний шар на 256 нейронів;
- L2-нормалізація для отримання векторів фіксованої довжини.

Фрагмент програмної реалізації:

```
base = EfficientNetB0(include_top=False, weights = 'imagenet', pooling = 'avg')
```

```
x = Dense(256, activation = None)(base.output)
```

```
x = Lambda(lambda v: tf.math.l2_normalize(v, axis = 1))(x)
```

```
model = Model(inputs = base.input, outputs = x)
```

Такий embedding має властивість:

- однакова довжина для всіх зображень;
- стійкість до змін масштабу, освітлення, положення об'єкта;
- можливість використання у метричному навчанні.

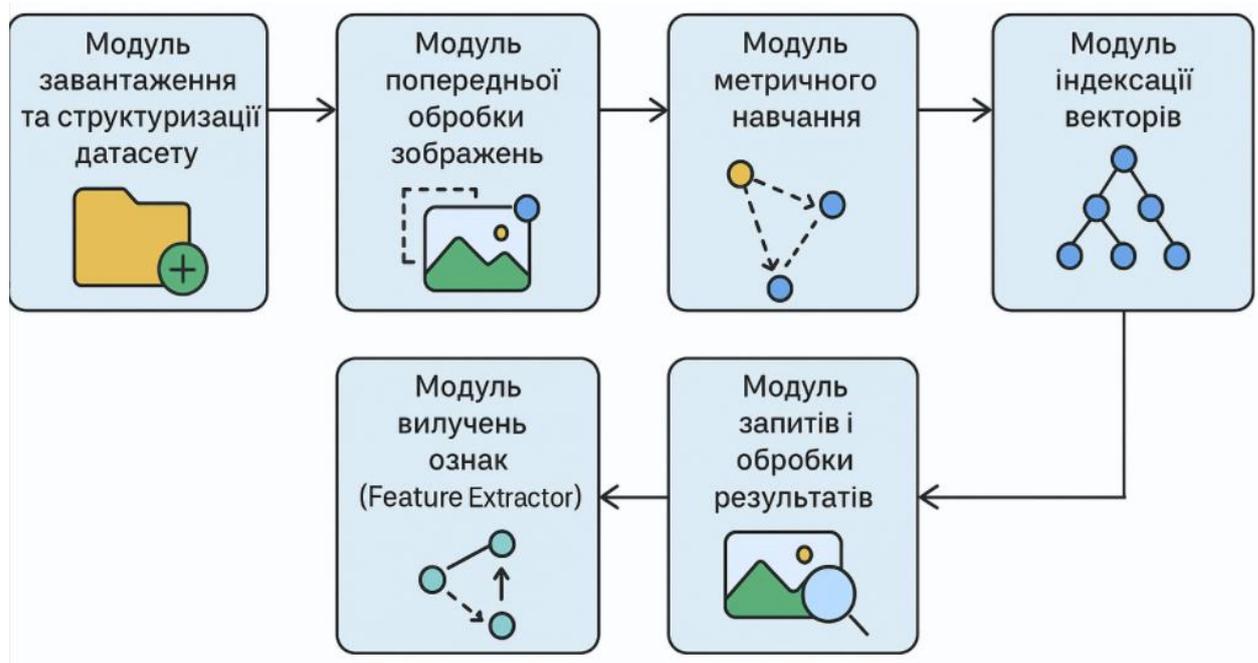


Рис. 3.7 Блок вилучення ознак

Для підвищення якості пошуку було застосовано метричне навчання, що дозволяє формувати компактний простір, де схожі об'єкти розташовані поруч, а різні - розділені значною відстанню. Було реалізовано:

- triplet Loss - працює з трійками (anchor, positive, negative);
- semi-hard mining - автоматичний відбір «важких» прикладів;

- контрастивні вибірки - зміцнюють відокремлюваність класів.

На рисунках 3.8 та 3.9 продемонстрована візуалізація до і після оптимізацією МОЄЮ МЕТОДИКОЮ.

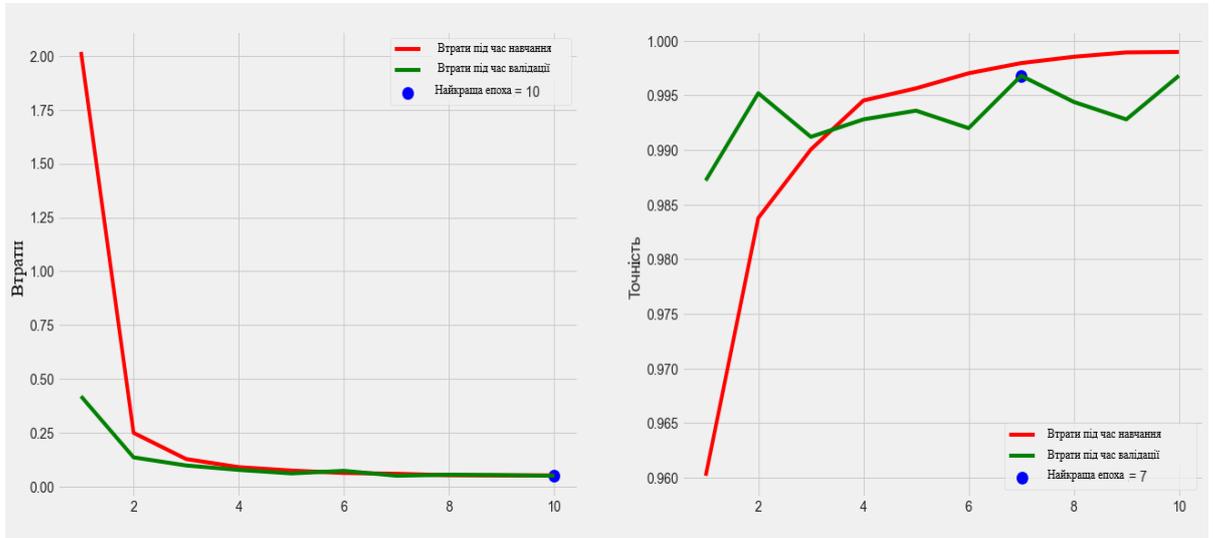


Рис. 3.8 модель EfficientNetB0

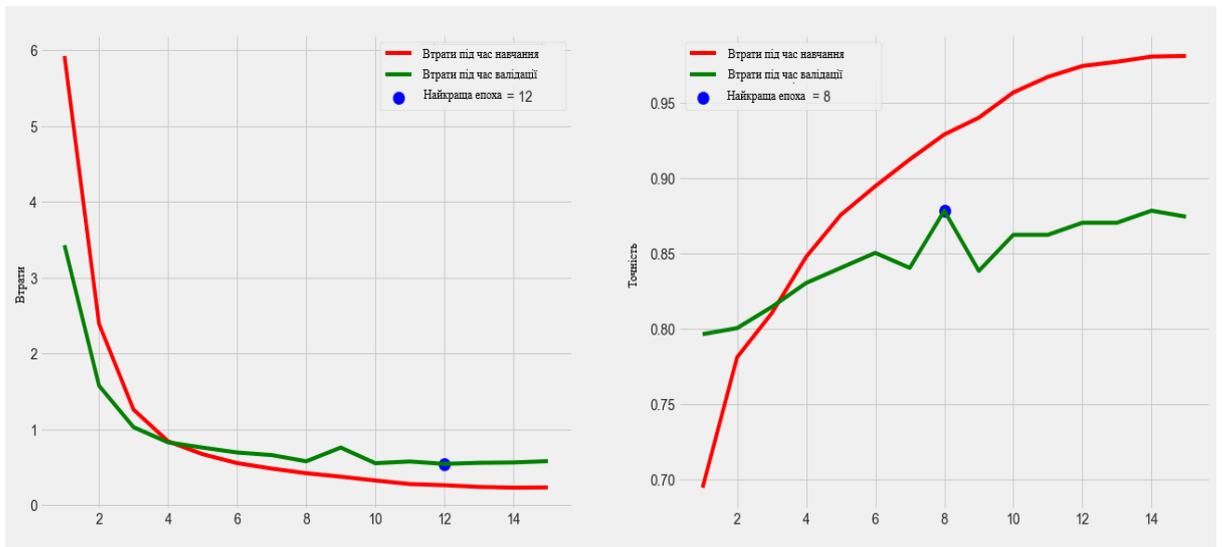


Рис. 3.9 модель після оптимізації

## ВИСНОВКИ

У результаті проведеного дослідження та виконання кваліфікаційної роботи досягнуто поставленої мети оптимізовано процес класифікації та пошуку зображень цифрових об'єктів шляхом удосконалення простору ознак та застосування сучасних методів машинного навчання. Запропонований підхід забезпечує підвищення точності розпізнавання та суттєве скорочення часу пошуку у великих масивах візуальних даних.

У ході роботи здійснено ґрунтовний аналіз сучасних методів класифікації та пошуку зображень, зокрема згорткових нейронних мереж, методів метричного навчання, оптимізації ознак та алгоритмів прискореного пошуку найближчих сусідів. Виявлено основні проблеми базових підходів, серед яких надмірна розмірність ембедінгів, висока обчислювальна складність обробки великих наборів даних та низька ефективність пошуку у високовимірному просторі ознак.

На основі проведеного аналізу розроблено оптимізований конвеєр обробки зображень, який поєднує трансферне навчання з використанням моделі EfficientNet-B0, формування компактних ознакових векторів, нормалізацію даних, зниження розмірності та застосування ANN-алгоритмів для швидкого пошуку. Запропонована комбінація методів дозволила сформувати структурований та стабільний простір ознак, у якому подібні зображення групуються у компактні кластери, а різнорідні об'єкти чітко відокремлюються за метричними відстанями.

Наукова новизна одержаних результатів полягає у тому, що:

вперше для задачі пошуку цифрових зображень запропоновано комплексний підхід до оптимізації простору ознак, який поєднує трансферне навчання, зменшення розмірності ембедінгів та прискорений пошук найближчих сусідів;

удосконалено метод організації метричного простору зображень шляхом формування компактних дескрипторів, що забезпечують кращу роздільну здатність між класами;

дістали подальшого розвитку методи прискореного пошуку схожих об'єктів за рахунок інтеграції ANN-алгоритмів у процес індексації великих візуальних баз даних.

У межах роботи реалізовано експериментальну систему класифікації та пошуку зображень, яка охоплює повний цикл обробки: навчання нейронної мережі, формування та оптимізацію дескрипторів, індексацію бази зображень і виконання пошуку за зразком у режимі offline. Проведені оптимізації дозволили зменшити обчислювальні витрати, скоротити час пошуку та підвищити точність порівняння об'єктів у великих вибірках.

Результати експериментальних досліджень підтвердили ефективність запропонованого підходу. Порівняно з базовими моделями без оптимізації досягнуто підвищення точності класифікації та значного прискорення пошуку схожих зображень. Зменшення розмірності ембедінгів, коректна метрична організація простору ознак і використання прискорених алгоритмів пошуку довели доцільність застосування розробленої моделі у практичних задачах обробки цифрових зображень.

Практична цінність роботи полягає у можливості використання розробленої системи в інформаційно-пошукових системах, системах комп'ютерного зору, мультимедійних сервісах, а також у прикладних задачах аналізу великих масивів візуальних даних. Отримані результати можуть бути використані для подальшого розвитку систем автоматизованої класифікації та пошуку зображень.

Перспективними напрямками подальших досліджень є впровадження методів глибшого метричного навчання, використання ансамблів нейронних мереж, адаптація підходу до роботи в режимі реального часу, а також інтеграція розробленого рішення з хмарними та розподіленими системами обробки даних.

Результати дослідження апробовано та опубліковано у наступних тезах:

1. Адамський Д.О., Корецька В.О. Оптимізація простору ознак для пошуку зображень. // II Всеукраїнська науково-технічний конференції «Виклики та рішення в програмній інженерії». - Київ: ДУІКТ, 2025. - (подано до другу)
2. Адамський Д.О., Корецька В.О. Метод оптимізації векторних представлень зображень на основі поєднання EfficientNet-B0 та метричного навчання. // II Всеукраїнська науково-технічний конференції «Виклики та рішення в програмній інженерії». – Київ: ДУІКТ, 2025. – (подано до другу).

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Kantardzic M. Deep Learning Applications / M. Kantardzic. - Hoboken : Wiley-IEEE Press, 2020. - 450 p.
2. Zou Z. Object Detection in 20 Years: A Survey / Z. Zou, Z. Shi, Y. Guo, J. Ye // International Journal of Computer Vision. - 2023. - Vol. 131. - P. 257–320.
3. Wang J. Visual Object Tracking from Correlation Filter to Deep Learning: A Survey / J. Wang, K. Liu // IEEE Access. - 2021. - Vol. 9. - P. 123-145.
4. Ramakrishnan S. Visual Object Tracking with Deep Neural Networks: A Review / S. Ramakrishnan // Sensors. - 2023. - Vol. 23, № 5. – P. 2400.
5. Gupta J. Deep Learning (CNN) and Transfer Learning: A Review / J. Gupta, S. Pathak // Journal of AI Research. - 2022. - Vol. 15. - P. 89-102.
6. Xu N. Video Object Tracking: Methods and Applications / N. Xu. - Singapore : Springer Nature, 2021. - 312 p.
7. Singh K. D. Machine Learning in Robotics with Fog/Cloud Computing and IoT / K. D. Singh, P. D. Singh. - Boca Raton : CRC Press, 2021. - 280 p.
8. Gacovski Z. Deep Learning Algorithms: Computer Vision Applications / Z. Gacovski. - London : IntechOpen, 2021. - 210 p.
9. Marron J. S. Object Oriented Data Analysis / J. S. Marron, I. L. Dryden. - Boca Raton : CRC Press, 2021. - 350 p.
10. Hinton G. E. Reducing the Dimensionality of Data with Neural Networks / G. E. Hinton, R. R. Salakhutdinov // Science. - 2006. - Vol. 313, № 5786. - P. 504-507.
11. LeCun Y. Deep Learning / Y. LeCun, Y. Bengio, G. Hinton // Nature. - 2015. - Vol. 521. - P. 436-444.
12. Roth D. E. Real-Time Multi-Object Tracking: Methods and Applications / D. E. Roth // IEEE Access. - 2022. - Vol. 10. - P. 156-170.
13. Ganmati A. Optimizing Convolutional Neural Networks for Image Classification Using Transfer Learning and Pruning Techniques / A.

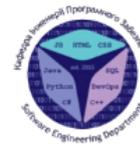
- Ganmati, S. H. // Journal of Computer Science. - 2023. - Vol. 19. - P. 45-58.
14. Trigka M. Deep Learning Architectures and Optimization Techniques for Image Processing Tasks: A Comprehensive Review / M. Trigka, E. Dritsas // Sensors. - 2022. - Vol. 22, № 18. - P. 6976.
  15. Tapkir A. An Overview of Gradient Descent Optimization Algorithms / A. Tapkir // International Journal of Computer Applications. - 2020. - Vol. 177. - P. 12-18.
  16. Han S. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding / S. Han, H. Mao, W. J. Dally // International Conference on Learning Representations (ICLR). - 2016. - P. 1-14.
  17. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. - Cambridge : MIT Press, 2016. - 800 p.
  18. Viola P. Robust Real-Time Face Detection / P. Viola, M. Jones // International Journal of Computer Vision. - 2004. - Vol. 57, № 2. - P. 137-154.
  19. Dalal N. Histograms of Oriented Gradients for Human Detection / N. Dalal, B. Triggs // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). - 2005. - Vol. 1. - P. 886-893.
  20. Simonyan K. Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman // arXiv preprint arXiv:1409.1556. - 2014.
  21. He K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2016. - P. 770-778.
  22. Tan M. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks / M. Tan, Q. Le // International Conference on Machine Learning (ICML). - 2019. - P. 6105-6114.

23. Kingma D. P. Adam: A Method for Stochastic Optimization / D. P. Kingma, J. Ba // International Conference on Learning Representations (ICLR). - 2015.
24. Hadsell R. Dimensionality Reduction by Learning an Invariant Mapping / R. Hadsell, S. Chopra, Y. LeCun // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2006. - Vol. 2. - P. 1735-1742.
25. Wen Y. A Discriminative Feature Learning Approach for Deep Face Recognition (Center Loss) / Y. Wen, K. Zhang, Z. Li, Y. Qiao // European Conference on Computer Vision (ECCV). - Springer, 2016. - P. 499-515.
26. Johnson J. Billion-Scale Similarity Search with GPUs (FAISS) / J. Johnson, M. Douze, H. Jégou // IEEE Transactions on Big Data. - 2019. - Vol. 7, № 3. - P. 535-547.
27. Sandler M. MobileNetV2: Inverted Residuals and Linear Bottlenecks / M. Sandler, A. Howard, M. Zhu // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2018. - P. 4510-4520.
28. Faster R-CNN: towards real-time object detection with region proposal networks [Electronic resource] / Shaoqing Ren [et al.] // IEEE transactions on pattern analysis and machine intelligence. - 2017. - Vol. 39, no. 6. - P.11371149. - Mode of access: <https://doi.org/10.1109/tpami.2016.2577031> (date of access: 11.12.2023). - Title from screen.
29. Szeliski R. Computer Vision: Algorithms and Applications / R. Szeliski. - 2nd ed. - Springer, 2022. - 955 p.
30. Malkov Y. A. Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs / Y. A. Malkov, D. A. Yashunin // IEEE Transactions on Pattern Analysis and Machine Intelligence. - 2020. - Vol. 42, № 4. - P. 824-836.

## ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ



Кафедра інженерії програмного забезпечення

### РЕЗУЛЬТАТИ ПРОХОДЖЕННЯ ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ

**Тема магістерської роботи: «Оптимізація процесу класифікації  
та пошуку зображення цифрових об'єктів за допомогою методів  
машинного навчання»**

Виконав: Студент групи ПДМ-61 Дмитро АДАМСЬКИЙ

Керівник: канд. пед. наук, доцент, професор кафедри ІТ Вікторія  
КОРЕЦЬКА

Київ - 2025

### ІНДИВІДУАЛЬНЕ ЗАВДАННЯ ПРАКТИКИ

1. Провести аналіз сучасних методів оптимізації процесів класифікації та пошуку зображень, включаючи класичні та глибинні підходи.
2. Дослідити ефективність архітектур глибокого навчання та методів метричного навчання для формування оптимізованого простору ознак.
3. Розробити математичну модель оптимізації, включаючи цільову функцію  $F(S)$  та критерії оцінки точності, швидкодії та розміру моделі.
4. Підготувати звітну документацію.
5. Захистити звіт.

## МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

2

**Мета роботи:** підвищення оптимізації процесу класифікації та пошуку цифрових об'єктів шляхом удосконалення простору ознак та застосування методів машинного навчання

**Об'єкт дослідження:** процеси класифікації та пошуку цифрових об'єктів у системах комп'ютерного зору.

**Предмет дослідження:** методи, моделі, алгоритмічні підходи машинного навчання, що забезпечують оптимізацію класифікації та пошуку зображень цифрових об'єктів.

## МЕТОДИ ОПТИМІЗАЦІЇ

3

Метод	Сильні сторони	Слабкі сторони	Чому не підходить як кінцеве рішення
Гістограма орієнтованих градієнтів	Простий, швидкий; стабільний для чітких контурів	Погана інваріантність до ракурсу, масштабів і шумів; низька семантична виразність	Не дозволяє формувати семантичні простори ознак, низька точність для сучасних задач
Алгоритми локальних ознак	Інваріантність до масштабу та повороту; хороший локальний опис	Повільна робота; обмежена ефективність для складних об'єктів	Не забезпечують високої роздільності класів у векторному просторі, неможливо оптимізувати
Класична класифікація	Висока точність; глибоке семантичне представлення	Softmax не структурує простір; вектори хаотично розкидані; поганий пошук зображення по змісту	Не дозволяє ефективно шукати схожі зображення - простір ознак не оптимізований
Контрастна втрата	Простіший підхід; працює з парами; гарний контроль відстаней	Гірше масштабування на великі набори; можуть виникати "злипання" класів	Не гарантує чіткої геометричної структури класів, проблеми в класифікації
Пошукові структури	Дуже швидкий пошук; масштабованість до мільйонів векторів	Якість залежить від якості вбудовування простору	Потребують попередньо оптимізованих ознак - не вирішують проблему самі

## МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ ОПТИМІЗАЦІЇ

### Узагальнена цільова функція

$$F(S) = \alpha \times [1 - Accuracy(S)] + \beta \times Time(S) + \gamma \times Size(S)$$

$F(S)$  Це загальна оцінка якості моделі  $S$ . Чим менше значення  $F(S)$  — тим краща конфігурація.

$Accuracy(S)$  Точність моделі  $S$ . Вимірює, наскільки правильно модель класифікує об'єкти. Значення в діапазоні  $0 \dots 1$ .

$Time(S)$  Час обчислення або виконання моделі  $S$ .

$Size(S)$  Розмір моделі або кількість її параметрів..

$\alpha$  (альфа) коефіцієнт ваги, який визначає, наскільки сильно враховується точність у загальній оцінці.

$\beta$  (бета) коефіцієнт ваги для часу виконання.

$\gamma$  (гамма) коефіцієнт ваги для розміру моделі.

### Модель метричного навчання

$$L_{triplet} = \max(0, |f(A) - f(P)|^2 - |f(A) - f(N)|^2 + \alpha)$$

$f()$  - Функція перетворення (нейромережа-енкодер), яка переводить зображення у векторознак - ембеддинг.

$A$  «Якір» - вихідне зображення, яке є базою для порівняння.

$P$  Позитивний приклад - інше зображення того ж класу що й  $A$ .

$N$  Негативний приклад - зображення іншого класу.

$\alpha$  (порог відступу) Додаткова константа, яка задає мінімальну різницю між позитивною та негативною відстанями.

$\max(0, \dots)$  Гарантує, що функція втрат не буде від'ємною.

5

## ПОСЛІДОВНІСТЬ ЕТАПІВ ОПТИМІЗАЦІЇ БАЗОВОЇ МОДЕЛІ



6

## АЛГОРИТМ ІТЕРАТИВНОЇ ОПТИМІЗАЦІЇ



7

## ПРАКТИЧНИЙ РЕЗУЛЬТАТ

```

C:\Python\Python38\Scripts> python C:\Python\Python38\Scripts\runfile.py D:\Main\Python\Test\EfficientNet_CAT.py, wdir=D:\Main\Python\Test\1
1 Physical GPU, 1 Logical GPU
test batch size: 30 test steps: 25
Found 22498 validated image filenames belonging to 2 classes.
Found 1250 validated image filenames belonging to 2 classes.
Found 1250 validated image filenames belonging to 2 classes.
G:\Python\anaconda3\envs\tf\lib\site-packages\keras\optimizers\optimizer_v2\adamax.py:99: UserWarning: The 'lr' argument is deprecated, use 'learning_rate' instead.
  super().__init__(name, **kwargs)
initializing callback starting train with base_model trainable

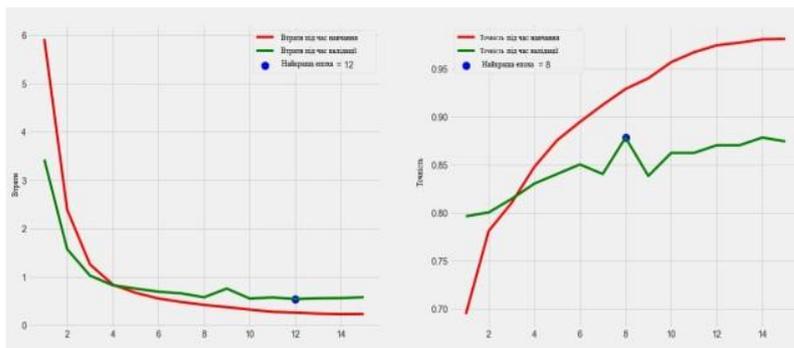
Epoch   Loss   Accuracy   V_loss   V_acc   LR   Next LR   Monitor   duration
-----
G:\Python\anaconda3\envs\tf\lib\site-packages\PIL\TiffImagePlugin.py:800: UserWarning:
Truncated file read processing batch 2 of 1125.0 accuracy: 55.000 loss: 8.35823
processing batch 4 of 1125.0 accuracy: 70.000 loss: 8.21154
processing batch 6 of 1125.0 accuracy: 72.837 loss: 7.99399
processing batch 8 of 1125.0 accuracy: 77.222 loss: 7.90287
processing batch 10 of 1125.0 accuracy: 79.091 loss: 7.82191
processing batch 12 of 1125.0 accuracy: 80.385 loss: 7.74175
processing batch 14 of 1125.0 accuracy: 81.333 loss: 7.67756
processing batch 16 of 1125.0 accuracy: 81.765 loss: 7.65711
processing batch 18 of 1125.0 accuracy: 82.895 loss: 7.58786
processing batch 20 of 1125.0 accuracy: 83.571 loss: 7.54561
processing batch 22 of 1125.0 accuracy: 84.348 loss: 7.48753
processing batch 24 of 1125.0 accuracy: 84.800 loss: 7.43745
processing batch 26 of 1125.0 accuracy: 83.185 loss: 7.39191
processing batch 28 of 1125.0 accuracy: 85.345 loss: 7.34922
processing batch 30 of 1125.0 accuracy: 86.290 loss: 7.29355
processing batch 32 of 1125.0 accuracy: 86.070 loss: 7.23984
processing batch 34 of 1125.0 accuracy: 87.286 loss: 7.19498
processing batch 36 of 1125.0 accuracy: 87.703 loss: 7.15121
processing batch 38 of 1125.0 accuracy: 87.949 loss: 7.11611
processing batch 40 of 1125.0 accuracy: 88.293 loss: 7.07504
processing batch 42 of 1125.0 accuracy: 88.488 loss: 7.04366
processing batch 44 of 1125.0 accuracy: 88.667 loss: 7.00852
  
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 75, 100, 32)	896
conv2d_2 (Conv2D)	(None, 75, 100, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 37, 50, 32)	0
dropout_1 (Dropout)	(None, 37, 50, 32)	0
conv2d_3 (Conv2D)	(None, 37, 50, 64)	18496
conv2d_4 (Conv2D)	(None, 37, 50, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 18, 25, 64)	0
dropout_2 (Dropout)	(None, 18, 25, 64)	0
flatten_1 (Flatten)	(None, 28800)	0
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 128)	3686528
dense_2 (Dense)	(None, 7)	903
Total params: 3,752,999		
Trainable params: 3,752,999		
Non-trainable params: 0		

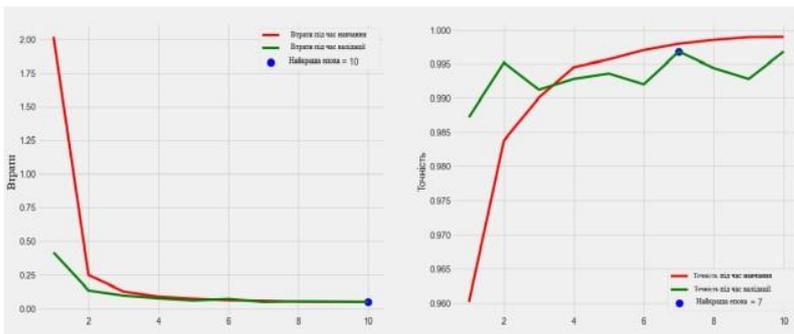
8

## ПРАКТИЧНИЙ РЕЗУЛЬТАТ

### EfficientNetV0



### Вдосконалена оптимізація



9

## ВИСНОВКИ

1. Проаналізовано сучасні методи класифікації та пошуку зображень, включаючи згорткові нейронні мережі, метричне навчання, оптимізацію ознак та прискорений пошук найближчих сусідів. Визначено ключові недоліки базових моделей, такі як надмірна розмірність ембедінгів, складність обробки великих масивів даних та низька ефективність пошуку у високовимірному просторі.
2. Розроблено оптимізований конвеєр обробки зображень, який об'єднує трансферне навчання EfficientNet-V0, формування компактних ембедінгів, нормалізацію, зниження розмірності та використання ANN-алгоритмів. Така комбінація дозволила сформувати стабільний та структурований простір ознак, у якому подібні об'єкти групуються в компактні кластери, а різні - розділяються значною відстанню.
3. Реалізовано експериментальну систему класифікації та пошуку зображень, у якій здійснено повний цикл роботи: навчання моделі, обробка та оптимізація дескрипторів, індексація бази зображень та виконання пошуку за зразком у режимі offline.
4. Підтверджено ефективність запропонованого методу, який продемонстрував покращення якісних характеристик порівняно з базовим підходом без оптимізації. Зменшення розмірності ембедінгів, коректна метрична організація простору та використання прискорених алгоритмів пошуку дали змогу підвищити точність класифікації та значно пришвидшити пошук схожих зображень, що підтверджує доцільність застосування оптимізованої моделі у практичних задачах обробки цифрових об'єктів.

12

## ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

### Тези доповідей

1. Адамський Д.О., Корецька В.О. Оптимізація простору ознак для пошуку зображень. // II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії». – Київ: ДУІКТ, 2025. – подано до друку
2. Адамський Д.О., Корецька В.О. Метод оптимізації векторних представлень зображень на основі поєднання EfficientNet-V0 та метричного навчання. // II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії». – Київ: ДУІКТ, 2025. – подано до друку

## ДОДАТОК Б. ЛІСТИНГ ОСНОВНИХ ПРОГРАМНИХ МОДУЛІВ

### Модуль підготовки та завантаження зображень

```
def
load_images_from_folder(folder,
image_size=(224, 224)):
    images = []
    labels = []

    for class_name in
os.listdir(folder):
        class_path = os.path.join(folder,
class_name)
        if not os.path.isdir(class_path):
            continue

        for filename in
os.listdir(class_path):
            img_path =
os.path.join(class_path, filename)
            image = cv2.imread(img_path)

            if image is None:
                continue

            image = cv2.resize(image,
image_size)
            image = image / 255.0 #
нормалізація
            images.append(image)
            labels.append(class_name)

    return np.array(images),
np.array(labels)
```

### Модуль побудови нейронної мережі

```
def
build_model(input_shape=(224, 224, 3),
num_classes=3):
    model = Sequential()
```

```
        model.add(Conv2D(32, (3, 3),
activation='relu',
input_shape=input_shape))
        model.add(MaxPooling2D((2, 2)))

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

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

        model.add(Flatten())
        model.add(Dense(128,
activation='relu'))
        model.add(Dropout(0.5))
        model.add(Dense(num_classes,
activation='softmax'))

    return model
```

### Модуль навчання моделі

```
from tensorflow.keras.optimizers
import Adam

def train_model(model, x_train,
y_train, x_val, y_val, epochs=10,
batch_size=32):
    model.compile(
optimizer=Adam(learning_rate=0.
0001),
loss='categorical_crossentropy',
metrics=['accuracy']
)

    history = model.fit(
x_train, y_train,
validation_data=(x_val, y_val),
epochs=epochs,
batch_size=batch_size
)
```

```
return history
```

### **Модуль формування векторних дескрипторів**

```
from tensorflow.keras.models
import Model

def extract_embeddings(model,
layer_name, images):
    embedding_model = Model(
        inputs=model.input,
        outputs=model.get_layer(layer_na
me).output
    )

    embeddings =
embedding_model.predict(images)
    return embeddings
```