

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління  
(повна назва)

Кафедра електронних обчислювальних машин  
(повна назва)

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

Рівень вищої освіти перший (бакалаврський)

Програмні засоби обробки зображень з використанням  
штучних нейронних мереж

(тема)

Виконав:

здобувач 4 року навчання,

групи КІУКІ-21-2

Максим КАВЕРІН

(власне ім'я, прізвище)

Спеціальність

123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма

Комп'ютерна інженерія

(повна назва освітньої програми)

Керівник: ас. Олександр РОМАНЮК

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ електронних обчислювальних машин \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 «Комп'ютерна інженерія» \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Комп'ютерна інженерія \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Каверіну Максиму Олександровичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Програмні засоби обробки зображень з використанням штучних нейронних мереж \_\_\_\_\_

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії \_\_\_\_\_ 17 червня 2025 р.

3. Вхідні дані до роботи \_\_\_\_\_

набори зображень: MNIST, Fashion-MNIST, CIFAR-10

Google Colab

розпізнавання зображень

програмні засоби

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

Аналіз предметної області

Штучні нейронні мережі в задачах обробки зображень

Програмна реалізація та аналіз результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 13 слайдів

---

---

---

---

---

---

---

---

---

---

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

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	26.05.2025–30.05.2025	
2	Огляд існуючих рішень та алгоритмів	31.05.2025–03.06.2025	
3	Вибір датасетів та архітектури мережі	04.06.2025–06.06.2025	
4	Вибір програмних засобів	07.06.2025–08.06.2025	
5	Програмна реалізація	09.06.2025–11.06.2025	
6	Аналіз отриманих результатів	12.06.2025–13.06.2025	
7	Оформлення записки	14.06.2025–16.06.2025	

Дата видачі завдання “ 26 ” травня 2025 р.

Здобувач

  
(підпис)

Керівник роботи

(підпис)

ас. Олександр РОМАНЮК

(посада, власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 65 с., 25 рис., 2 дод., 7 джерел.

ОБРОБКА ЗОБРАЖЕНЬ, ШТУЧНА НЕЙРОННА МЕРЕЖА, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ, ГЛИБОКЕ НАВЧАННЯ, MNIST, FASHION-MNIST, CIFAR-10, GOOGLE COLAB, TENSORFLOW, KERAS, СЕГМЕНТАЦІЯ, АВТОЕНКОДЕР.

Метою кваліфікаційної роботи є розробка, реалізація та експериментальне дослідження програмних засобів обробки зображень на основі штучних нейронних мереж у середовищі Google Colab для підвищення точності автоматичного розпізнавання візуальних об'єктів.

У ході виконання кваліфікаційної роботи здійснено теоретичний аналіз архітектур сучасних нейромереж, включаючи LeNet, AlexNet, VGG та ResNet. Розглянуто базові принципи роботи автоенкодерів та застосування нейронних мереж у задачах сегментації. Проведено огляд найпоширеніших бібліотек для реалізації моделей машинного навчання: TensorFlow, Keras і PyTorch, а також обґрунтовано вибір середовища Google Colab для реалізації програмного експерименту. У практичній частині було реалізовано програмну систему, що здійснює попередню обробку зображень навчання CNN-моделі на трьох різних датасетах: MNIST, Fashion-MNIST та CIFAR-10. На основі результатів експериментів здійснено порівняльну оцінку ефективності нейронної мережі на зображеннях різної складності.

## ABSTRACT

Bachelor's thesis: 65 pages, 25 figures, 2 appendices, 7 sources.

IMAGE PROCESSING, ARTIFICIAL NEURAL NETWORK, CONVOLUTIONAL NEURAL NETWORK, IMAGE CLASSIFICATION, MACHINE LEARNING, DEEP LEARNING, MNIST, FASHION-MNIST, CIFAR-10, GOOGLE COLAB, TENSORFLOW, KERAS, SEGMENTATION, AUTOENCODER.

The major goal of this thesis is the development, implementation, and experimental evaluation of image processing software based on artificial neural networks in the Google Colab environment, aimed at improving the accuracy of automatic visual object recognition.

In order to a theoretical analysis of modern neural network architectures was carried out, including LeNet, AlexNet, VGG, and ResNet. Fundamental principles of autoencoders and their application to image segmentation tasks were explored. A comprehensive review of the most widely used machine learning libraries: TensorFlow, Keras, and PyTorch was conducted, along with a substantiated rationale for the use of Google Colab as the development and experimentation platform.

In the practical part of the project, a software system was implemented to perform image preprocessing and to train convolutional neural networks (CNNs) using three distinct datasets: MNIST, Fashion-MNIST, and CIFAR-10. Based on the experimental results, a comparative evaluation of the network's performance across different image complexities was conducted.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	8
ВСТУП .....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	12
1.1 Методи розпізнавання зображень .....	12
1.1.1 Метод Scale-Invariant Feature Transform .....	13
1.1.2 Метод Speeded-Up Robust Features .....	16
1.1.3 Метод Oriented FAST and Rotated BRIEF .....	20
1.1.4 Binary Robust Independent Elementary Features .....	21
1.2 Порівняльний аналіз розглянутих методів .....	23
1.3 Методи попередньої обробки зображень .....	25
2 ШТУЧНІ НЕЙРОННІ МЕРЕЖІ В ЗАДАЧАХ ОБРОБКИ ЗОБРАЖЕНЬ .....	27
2.1 Основи штучних нейронних мереж .....	29
2.2 Архітектури CNN .....	30
2.3 Автоенкодера та сегментація.....	32
2.4 Бібліотеки реалізації нейронних мереж.....	34
2.5 Обґрунтування вибору ШНМ для поставленого завдання .....	36
3 РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ .....	38
3.1 Вибір інструментів і середовища розробки.....	38
3.2 Обґрунтування вибору датасетів .....	39
3.3 Реалізація та аналіз результатів .....	41
ВИСНОВКИ.....	51
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	53
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	54
ДОДАТОК Б Програмний код .....	62
Б.1 Лістинг коду .....	62

Б.1.1 Імпорт бібліотек, початкові налаштування та завантаження MNIST .....	62
Б.1.2 Візуалізація прикладів.....	62
Б.1.3 Побудова CNN для MNIST .....	62
Б.1.4 Навчання моделі та візуалізація процесу навчання .....	63
Б.1.5 Робота з датасет Fashion-MNIST .....	63
Б.1.6 Робота з датасет CIFAR-10 .....	64

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AA – автоенкодер

ЗНМ – згорткова нейронна мережа

КНМ – класична нейронна мережа

НМ – нейронна мережа

ППЗ – прикладне програмне забезпечення

ШНМ – штучна нейронна мережа

CNN – згорткова нейронна мережа

CIFAR-10 – набір зображень із 10 класами

GPU – графічний процесор

Keras – високорівнева бібліотека для глибокого навчання на Python

MNIST – база зображень рукописних цифр

ReLU – випрямлена лінійна функція активації

ResNet – Residual Network,

SIFT – Scale-Invariant Feature Transform

SURF – Speeded-Up Robust Features

TF – TensorFlow

VGG – Visual Geometry Group

## ВСТУП

У сучасних умовах стрімкого розвитку цифрових технологій обробка зображень відіграє ключову роль у багатьох прикладних сферах – від медичної діагностики й систем відеоспостереження до автономного водіння, промислової автоматизації та цифрової ідентифікації. Традиційні алгоритми обробки зображень базуються на жорстко заданих правилах і часто не здатні адаптуватися до складних, варіативних та зашумлених вхідних даних. Це істотно обмежує їх застосування в реальних умовах.

З появою та розвитком штучних нейронних мереж (ШНМ) відбувся прорив у сфері комп'ютерного зору. Зокрема, згорткові нейронні мережі (CNN) довели свою здатність ефективно розпізнавати візуальні патерни на зображеннях різної складності. Використання глибокого навчання дозволяє автоматизовано виділяти суттєві ознаки, знижуючи потребу в ручному конструюванні ознак, і досягати високих результатів у задачах класифікації, сегментації та генерації зображень.

Цифрова обробка зображень зосереджена переважно на поліпшенні їхньої якості, тоді як концепція комп'ютерного зору передбачає глибший рівень аналізу, орієнтований на автоматичне витягування смислових характеристик із візуальних даних. Одним із пріоритетних напрямів досліджень у цій галузі є розв'язання задач, пов'язаних з порівнянням візуальних об'єктів і виявленням зображень, що належать до однієї категорії або класу. Значна увага в наукових та прикладних розробках приділяється використанню алгоритмів, які дозволяють ефективно аналізувати ознаки зображень і виявляти між ними подібність за допомогою обчислюваних дескрипторів.

Застосування технологій комп'ютерного зору охоплює широке коло сфер – від систем технічного контролю та промислової автоматизації до інтелектуальних механізмів ідентифікації та моніторингу в транспортній,

виробничій чи безпековій інфраструктурі. Особливої актуальності набувають інструменти, що здатні автоматично класифікувати зображення та знаходити в них релевантні об'єкти, що є критично важливим у системах сортування, контролю якості й навігації. Реалізація таких рішень потребує не лише обізнаності в методах машинного зору, а й глибоких знань у суміжних галузях інформаційних технологій, включаючи обробку даних, архітектуру комп'ютерних систем і побудову розподілених обчислювальних платформ.

Розробка та практична реалізація програмних засобів на базі CNN у середовищі Google Colab дає змогу створити ефективний інструмент для аналізу зображень, доступний через хмарну інфраструктуру. У рамках цієї кваліфікаційної роботи реалізовано обробку трьох відкритих наборів даних – MNIST, Fashion-MNIST та CIFAR-10 – із застосуванням спеціалізованих архітектур CNN. Для кожного з наборів виконано візуалізацію даних, побудову та навчання моделі, аналіз ефективності та порівняння результатів.

Метою кваліфікаційної роботи є розробка, реалізація та експериментальне дослідження програмних засобів обробки зображень на основі штучних нейронних мереж у середовищі Google Colab для підвищення точності автоматичного розпізнавання візуальних об'єктів.

Задачі:

- провести аналіз сучасних підходів до обробки зображень із використанням штучних нейронних мереж;
- обґрунтувати вибір архітектури CNN для вирішення задачі класифікації зображень;
- вибрати та підготувати три репрезентативні набори зображень: MNIST, Fashion-MNIST та CIFAR-10;
- реалізувати програмні засоби попередньої обробки та візуалізації зображень у Google Colab;
- побудувати та навчити три моделі CNN відповідно до кожного з обраних датасетів;
- виконати візуалізацію результатів, включаючи графіки навчання та

матриці плутанини;

- провести порівняльний аналіз ефективності моделей за точністю, швидкістю та візуальними характеристиками.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Методи розпізнавання зображень

У сфері машинного навчання, розпізнавання образів та цифрової обробки зображень важливе місце посідає процес трансформації первинних даних у більш інформативне, компактне й придатне для подальшого аналізу представлення. Цей процес передбачає побудову ознак, які не лише зберігають ключову інформацію про об'єкти дослідження, а й знижують надмірність і складність вхідного простору, сприяючи ефективнішому навчанню моделей і полегшенню інтерпретації результатів. Зазначена процедура часто супроводжується зменшенням розмірності, що дозволяє усунути зайві параметри та фокусуватися на найбільш значущих характеристиках даних.

У випадках, коли обсяг інформації є надто великим або структурно перевантаженим, застосовують методи виділення ознак, які забезпечують узагальнене представлення збережених властивостей для подальшого використання в задачах класифікації, розпізнавання або пошуку.

Потреба у стислому, але змістовному відображенні даних виникає в різних галузях, зокрема в контексті зберігання великих масивів інформації, прискорення обчислень або підвищення ефективності порівняння візуальних об'єктів. Особливо це актуально для систем, які функціонують в умовах обмежених ресурсів або потребують оперативної обробки великої кількості зображень. Водночас універсальних рішень щодо порівняння візуальних даних на сьогодні не існує.

Ефективність таких алгоритмів значною мірою залежить від специфіки задачі, тому для досягнення прийнятної точності зазвичай розробляють спеціалізовані методики з можливістю налаштування параметрів відповідно до характеру вхідних даних та умов експлуатації.

### 1.1.1 Метод Scale-Invariant Feature Transform

Один із найвідоміших алгоритмів для виявлення та опису ключових точок на зображеннях – це метод Scale-Invariant Feature Transform (SIFT) (рисунок 1.1), запропонований Д. Лоу в 2004 році в межах досліджень, проведених в Університеті Британської Колумбії. Алгоритм набув широкого визнання завдяки своїй здатності забезпечувати інваріантність до масштабних змін зображення, а також до його поворотів. Ця властивість дозволяє використовувати метод у завданнях, де порівнювані об'єкти можуть бути представлені в різних умовах зйомки, зокрема при зміні відстані або орієнтації. Через наявність патентного захисту реалізація цього методу включена в спеціальний модуль OpenCV з обмеженим доступом [1].

Основу роботи методу становить побудова багатомасштабного представлення зображення, що ґрунтується на згортці зображення з гауссовим ядром на різних масштабах. Це дозволяє сформувати простір масштабу, в межах якого відбувається ідентифікація характерних точок. Зображення обробляються шляхом розбиття на октави, причому розмір зображення в кожній наступній октаві зменшується вдвічі відносно попередньої. Кожна октава, у свою чергу, містить кілька рівнів, сформованих шляхом послідовного розмиття (рисунок 1.2), що дає змогу виділити стійкі до шуму особливості.

З метою виявлення потенційно інформативних областей, побудову гауссових масштабів доповнює операція вирахування зображень, відома як Difference of Gaussians (DoG) (рисунок 1.3). Ці зображення ефективно використовуються для виявлення ключових точок, що локалізуються на краях, кутах або текстурних переходах. Пошук таких точок виконується незалежно в кожній октаві побудованої гауссової піраміди, що забезпечує багаторівневий аналіз зображення та високу стійкість до змін вхідного сигналу.

Алгоритм SIFT характеризується не лише високою локалізаційною

точністю, але й здатністю масштабуватися до великих обсягів даних завдяки ефективному механізму зіставлення ознак. Він дозволяє генерувати велику кількість дескрипторів навіть для невеликих об'єктів, зберігаючи при цьому здатність до узгодження з великими базами зображень. Гнучкість методу також дозволяє його розширення для роботи з різними типами ознак, що підвищує надійність систем комп'ютерного зору при розв'язанні широкого спектра прикладних задач.

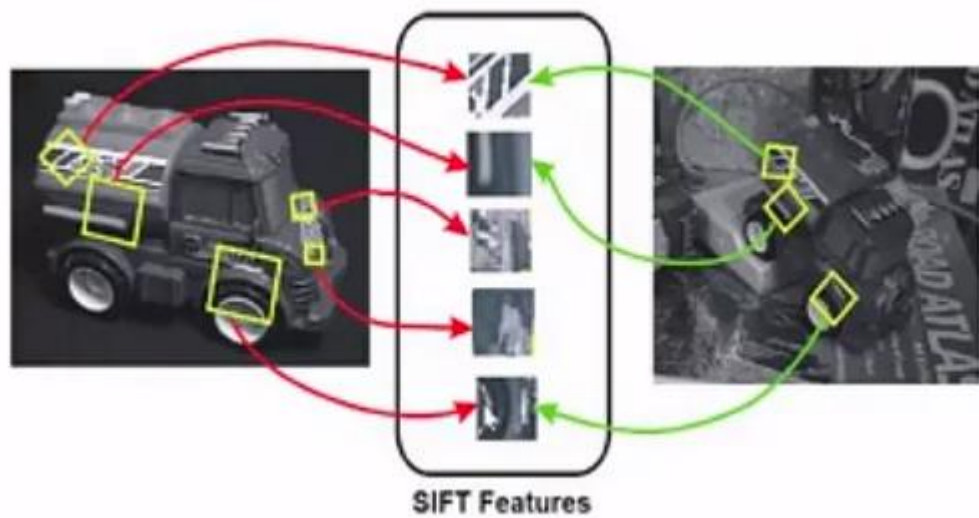


Рисунок 1.1 – Метод «SIFT»



Рисунок 1.2 – Приклад розмиття та масштабування зображення

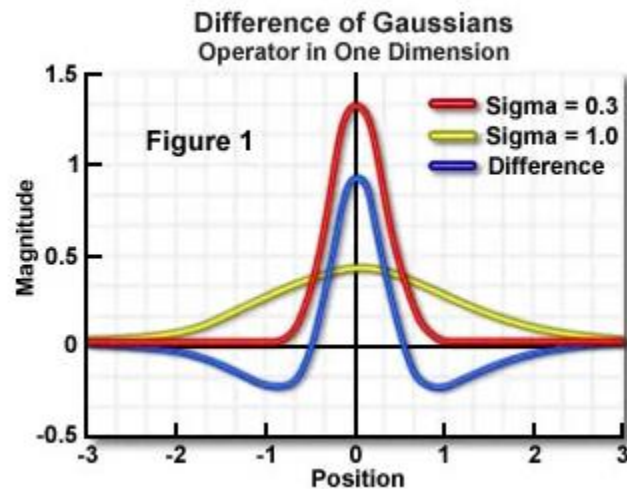


Рисунок 1.3 – Приклад Difference of Gaussians

Після побудови гауссових масштабів алгоритм SIFT виконує процедуру виявлення локальних екстремумів на основі наближення лапласіану зображення, що забезпечує інваріантність до масштабних змін. Кожен піксель аналізується в контексті його просторового оточення: значення інтенсивності порівнюються з вісімкою сусідніх пікселів у поточному масштабі, а також із дев'ятьма точками у вищому та дев'ятьма у нижчому масштабному рівні. Такий підхід дозволяє здійснити 26 незалежних перевірок на предмет екстремальності. Якщо піксель виявляється локальним максимумом або мінімумом у цьому тривимірному масштабно-просторовому середовищі, він позначається як потенційна ключова точка, тобто така, що найбільш повно відображає особливості зображення саме в цьому масштабному контексті.

У результаті цього етапу формується велика кількість кандидатів у ключові точки. Втім, значна їх частина може мати недостатню інформативність, зокрема у випадках, коли вони розташовані вздовж меж або демонструють низький рівень контрастності. Подібні особливості, як правило, не сприяють якісному порівнянню зображень, тому підлягають відсіву. Фільтрація виконується на основі аналізу геометричних властивостей околиці пікселя та величини градієнта, що нагадує підхід, застосований у

детекторі Харріса для придушення неінформативних ознак. Якщо контрастність точки не перевищує встановленого порогу, вона не включається до фінального набору.

Після етапу фільтрації залишаються стійкі ключові точки, для яких відомі координати, масштаб, у якому вони були виявлені, та орієнтація. Знаючи масштаб, визначають область навколо кожної ключової точки, де обчислюють напрямки градієнтів. На цій основі формується гістограма орієнтацій, яка охоплює повний кутовий діапазон у 360 градусів та складається зі 36 інтервалів. Найвищий пік гістограми визначає домінуючу орієнтацію ключової точки. Якщо існують інші піки, значення яких перевищують 80% від максимального, вони також враховуються, що призводить до створення додаткових точок з однаковим положенням і масштабом, але різними орієнтаціями. Це підвищує стійкість подальшого зіставлення ознак.

На цьому етапі кожна ключова точка характеризується повним набором параметрів: просторовим положенням, масштабом та орієнтацією. Далі здійснюється побудова дескриптора – векторного представлення локальної області зображення, яке повинно бути максимально унікальним та інваріантним до зовнішніх факторів, зокрема змін точки зору або умов освітлення. Останній крок передбачає зіставлення дескрипторів між різними зображеннями шляхом пошуку найближчих сусідів у просторі ознак, що дозволяє ідентифікувати спільні ключові точки та виконати коректне порівняння візуальних об'єктів.

### 1.1.2 Метод Speeded-Up Robust Features

Метод SURF був запропонований у 2006 році як відповідь на обмеження алгоритму SIFT, зокрема щодо його обчислювальної складності. Основна мета полягала в тому, щоби зберегти стійкість до змін масштабу, освітлення та обертання, але забезпечити вищу швидкодію. SURF базується

на гесіанських матрицях, які використовуються для виявлення ключових точок у багатомасштабному просторі. На відміну від SIFT, який застосовує згортки з гауссовими ядрами, у SURF використовуються апроксимації цих згорток за допомогою інтегральних зображень і прямокутних фільтрів, що істотно знижує обчислювальні витрати (рисунок 1.4).

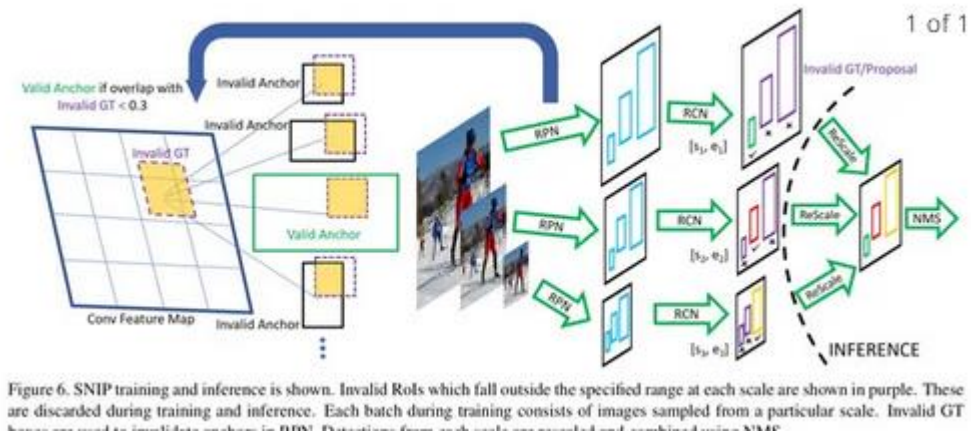


Figure 6. SNIP training and inference is shown. Invalid RoIs which fall outside the specified range at each scale are shown in purple. These are discarded during training and inference. Each batch during training consists of images sampled from a particular scale. Invalid GT

Рисунок 1.4 – Визначення маркеру за допомогою SURF

Орієнтація ключових точок визначається за допомогою обчислення гаусс-важених векторів градієнтів у межах локальної області, розташованої навколо точки, що дозволяє зберігати інваріантність до повороту. Створення дескриптора базується на аналізі розподілу інтенсивностей градієнтів у сітці з підрозділів, що формують вектор ознак. Кожен такий підрозділ представляється значеннями сум градієнтів і їх абсолютних значень по осях X та Y. Таким чином, формується вектор з 64 елементів (у базовій реалізації), що дозволяє досягати балансу між описовістю та ефективністю.

SURF демонструє високу швидкодію, яка особливо помітна при роботі з великими обсягами зображень або у реальному часі. Його дескриптори є менш чутливими до шумів і змін освітлення, а загальна архітектура забезпечує високу стійкість при відносно невеликих витратах ресурсів. Водночас варто зауважити, що, як і у випадку з SIFT, алгоритм SURF також перебуває під патентним захистом і доступний в OpenCV лише через модуль non-free.

Аналіз зображень у масштабному просторі традиційно реалізується шляхом побудови зображень у вигляді пірамід, де кожен рівень відповідає іншому ступеню згладжування та зменшення роздільної здатності. У класичних підходах для цього використовується поетапне застосування гауссових фільтрів із наступним субвибірковим масштабуванням. У випадку алгоритму SURF запропоновано принципово інший підхід до побудови масштабного простору: замість зменшення розміру зображення з кожною новою октавою, відбувається збільшення розміру фільтра, який накладається безпосередньо на вихідне зображення. Це забезпечується застосуванням інтегральних зображень та віконних фільтрів, що дозволяють виконувати згортку великих фільтрів на початковому зображенні без додаткових ітерацій та з постійною швидкістю, незалежно від розміру ядра.

У такій реалізації масштабного аналізу зміна масштабу досягається не через поетапне зменшення розміру зображення, а через поступове збільшення розміру згорткового фільтра – від  $9 \times 9$  до  $15 \times 15$ ,  $21 \times 21$ ,  $27 \times 27$  тощо. Кожна наступна октава пов'язана з подвоєнням розміру фільтра та відповідним збільшенням інтервалів вибірки при виявленні ключових точок. Такий підхід дозволяє ефективно масштабувати обчислення, зберігаючи при цьому стабільність витрат ресурсів. Для локалізації точок інтересу як у просторі координат, так і в масштабі, використовується тривимірна процедура пригнічення немаксимумів у вікні розміром  $3 \times 3 \times 3$ .

Створення дескриптора в алгоритмі SURF передбачає два послідовні етапи. На першому з них визначається стабільна орієнтація ключової точки. Це здійснюється шляхом аналізу розподілу векторів градієнтів у круговій області, розташованій навколо виявленої точки інтересу. Встановлення орієнтації необхідне для забезпечення інваріантності дескриптора до повороту зображення. Після фіксації орієнтації будується квадратна область, вирівняна відповідно до обраного напрямку, в межах якої відбувається формування самого дескриптора на основі локального розподілу ознак. Такий підхід дозволяє забезпечити високу стійкість ознак до геометричних

трансформацій і змін умов зйомки, що є критично важливим у прикладних задачах комп'ютерного зору (рисунок 1.5).

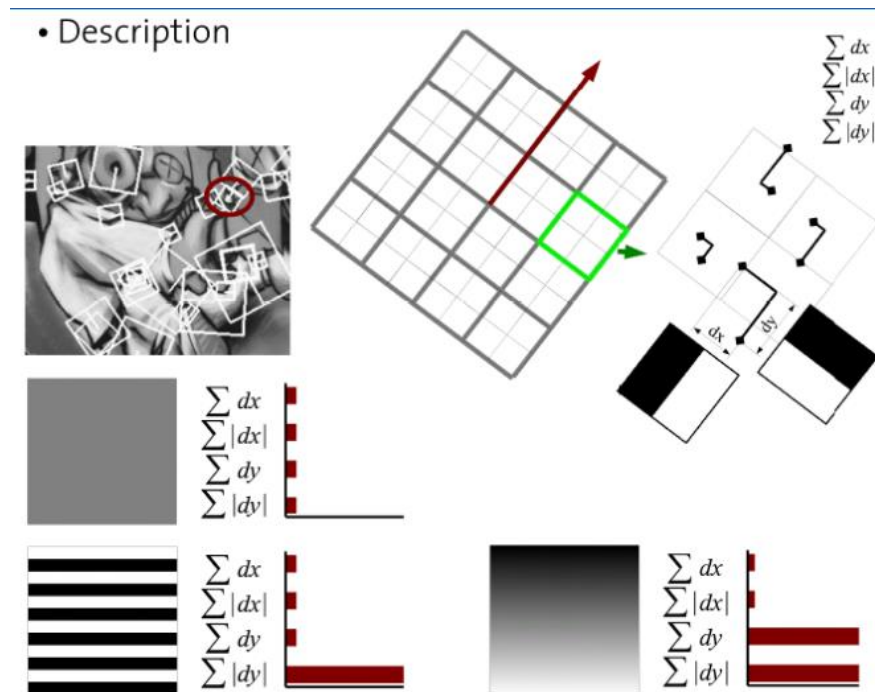


Рисунок 1.5 – Приклад роботи дескриптору

Після встановлення орієнтації та вирівнювання локальної області відповідно до неї, наступним етапом побудови дескриптора в алгоритмі SURF є просторове розбиття вибраної області на менші квадратні підрегіони. Така сіткова структура дозволяє детальніше охопити локальні особливості зображення навколо ключової точки та забезпечити інваріантність до малих геометричних деформацій. У межах кожного підрегіону на регулярно розміщених точках виконується обчислення фільтраційних відповідей, зокрема застосовуються вейвлет-фільтри Хаара в горизонтальному ( $dx$ ) та вертикальному ( $dy$ ) напрямках. Ці відповіді відображають зміну інтенсивності зображення по осях і служать основою для побудови дескриптора.

Для підвищення стійкості до похибок локалізації та незначних варіацій структури зображення вейвлет-відповіді зважуються гауссовою функцією, центр якої співпадає з координатами ключової точки. Таке зважування дозволяє надати більшу значущість центральним частинам області та

зменшити вплив периферійних шумів. Після цього значення  $dx$  та  $dy$  підсумовуються в межах кожного субрегіону, формуючи основні складові векторного представлення ознаки. Додатково обчислюються суми абсолютних значень відповідей  $|dx|$  і  $|dy|$ , що дозволяє включити інформацію про полярність зміни інтенсивності, тобто напрямки контрасту, незалежно від знаку.

У результаті для кожного субрегіону формується чотиривимірний вектор, що характеризує локальну структуру інтенсивності. Сукупність таких векторів, отриманих з усіх підрегіонів, агрегується в єдиний дескриптор, який описує ключову точку з урахуванням її просторової, масштабної та орієнтаційної специфіки. Така побудова дозволяє досягти високої дискримінаційної здатності при порівнянні зображень, зберігаючи водночас обчислювальну ефективність, яка є ключовою перевагою алгоритму SURF.

### 1.1.3 Метод Oriented FAST and Rotated BRIEF

Алгоритм ORB (рисунок 1.6) був розроблений як відкритий, вільно доступний метод для ефективного виявлення та опису ключових точок, який поєднує переваги двох існуючих рішень – швидкодії FAST і компактності BRIEF, доповнюючи їх інваріантністю до повороту. У першому етапі ORB використовує покращений варіант FAST для детектування кутів, після чого до кожної точки застосовується обчислення орієнтації за допомогою аналізу векторів інтенсивності у локальній області.

Для побудови дескриптора ORB застосовується модифікована версія BRIEF, у якій бінарні тести виконуються з урахуванням обертання області навколо ключової точки відповідно до знайденої орієнтації. Це дозволяє забезпечити стійкість дескриптора до поворотів зображення, що робить його придатним для використання в задачах реального часу. Крім того, застосовується механізм відбору найбільш інформативних бінарних тестів на основі дисперсійного аналізу, що підвищує дискримінаційну здатність

дескрипторів.

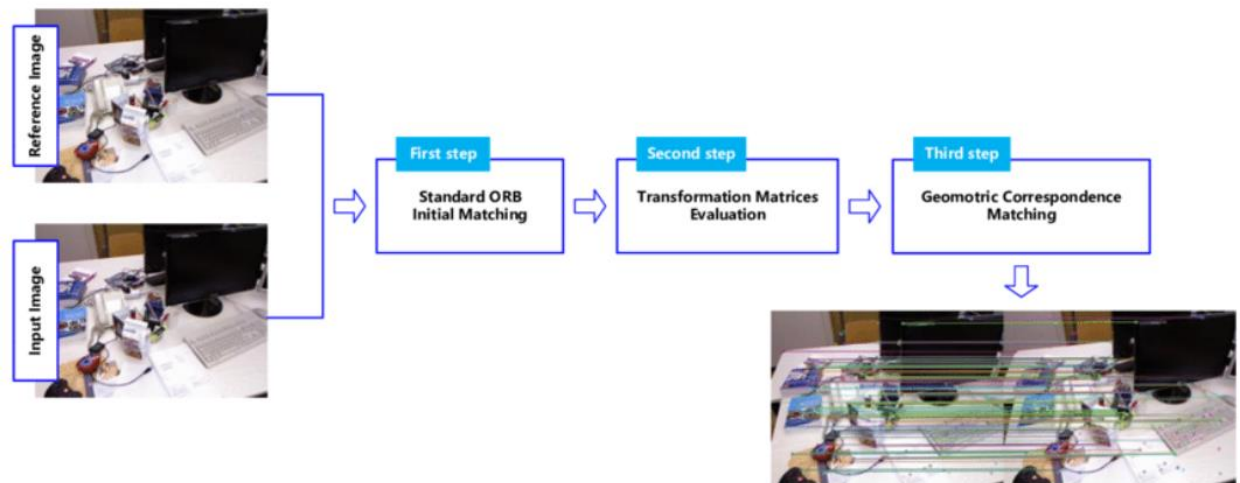


Рисунок 1.6 – Oriented FAST and Rotated BRIEF

ORB є повністю безкоштовним, не патентованим алгоритмом, що дозволяє вільно використовувати його в будь-яких проєктах. Він демонструє конкурентні результати порівняно з SIFT та SURF, забезпечуючи високу швидкість обробки та адекватну точність, особливо в умовах обмежених ресурсів. Алгоритм став однією з базових технологій у практичних застосуваннях комп'ютерного зору, зокрема в мобільних пристроях, вбудованих системах та інтернеті речей.

#### 1.1.4 Binary Robust Independent Elementary Features

Метод BRIEF(рисунок 1.7) є спрощеною альтернативою класичних дескрипторів ознак, запропонованою як високоефективне рішення для задач, що вимагають максимальної швидкості. Його концепція полягає в тому, щоб замінити багатовимірні дескриптори, засновані на градієнтах або гістограмах, на компактні бінарні вектори, що дозволяє істотно скоротити обсяг обчислень і пам'яті. На відміну від SIFT або SURF, BRIEF самостійно не виконує виявлення ключових точок, а лише генерує дескриптори для точок, отриманих іншим методом (наприклад, FAST, Harris).

Процес побудови дескриптора BRIEF включає вибір фіксованого набору пар пікселів у межах локальної області навколо ключової точки. Кожна пара порівнюється за інтенсивністю, і якщо інтенсивність у першому пікселі перевищує інтенсивність другого, у відповідному елементі дескриптора записується 1, інакше – 0. У результаті формується бітовий рядок, який слугує унікальним представленням локального візуального шаблону.

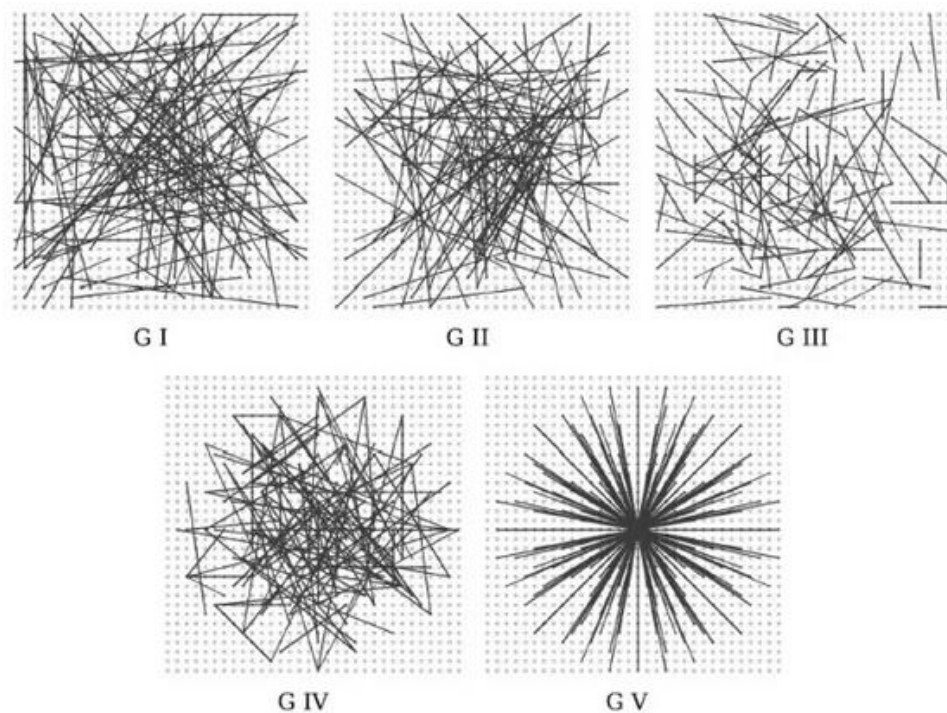


Рисунок 1.7 – Binary Robust Independent Elementary Features

Основними перевагами BRIEF є виняткова швидкодія, компактність дескрипторів і висока ефективність при порівнянні ознак, оскільки використовуються прості операції порівняння та обчислення метрики Хемінга. Водночас BRIEF не забезпечує інваріантності до повороту зображення та зміни масштабу, що обмежує його застосування у складних умовах або при аналізі різнопланових сцен.

## 1.2 Порівняльний аналіз розглянутих методів

У контексті задач комп'ютерного зору, пов'язаних із виявленням і описом ключових точок, розробка ефективних та надійних алгоритмів порівняння зображень є одним із ключових напрямів. Метод SIFT, запропонований Д. Лоу, довгий час вважався стандартом у цій галузі завдяки своїй стійкості до змін масштабу, обертання та частково – до освітлення. Його перевага полягає у глибокому багаторівневому аналізі зображення, який включає побудову масштабного простору, локалізацію екстремумів, визначення орієнтації та формування дескриптора на основі градієнтів. Такий підхід забезпечує високу точність при зіставленні зображень, однак має значні обчислювальні витрати, що обмежує його застосування в режимі реального часу та в системах із обмеженими ресурсами. Крім того, патентні обмеження на використання алгоритму обмежують його широке поширення в комерційних продуктах.

У відповідь на ці недоліки було запропоновано метод SURF, який зберігає основні концептуальні переваги SIFT, але використовує апроксимацію гауссових фільтрів через інтегральні зображення і вейвлети Хаара. Це дозволяє значно прискорити обчислення та забезпечити лінійну масштабованість при аналізі великих зображень. Водночас SURF демонструє подібну стійкість до змін масштабу та орієнтації, а його дескриптори забезпечують достатній рівень дискримінаційної здатності. Проте, як і SIFT, метод залишається патентованим, що обмежує його вільне використання в низці застосунків.

Іншим підходом до побудови дескрипторів стала концепція BRIEF, яка орієнтована на спрощення процесу шляхом використання бінарного представлення локальних ознак. Метод базується на простих порівняннях інтенсивностей піксельних пар у межах фіксованої області навколо ключової точки. Такий підхід забезпечує надзвичайно високу швидкодію та дозволяє суттєво зменшити обсяг пам'яті, необхідної для зберігання дескрипторів.

Однак BRIEF не враховує інформацію про орієнтацію або масштаб, що істотно знижує його стійкість до геометричних трансформацій, зокрема до повороту зображення. Через це застосування BRIEF обмежується задачами, де умови зйомки стабільні або контрольовані.

Метод	Інваріантність до масштабу	Інваріантність до обертання	Тип дескриптора	Швидкодія
SIFT	Так	Так	Вектор на основі градієнтів	Низька
SURF	Так	Так	Гаусс-зважені вейвлети Хаара	Середня
BRIEF	Ні	Ні	Бінарні порівняння інтенсивності	Висока
ORB	Частково	Так	Бінарні з орієнтацією	Висока
Метод	Використання в реальному часі	Патентні обмеження	Надійність у складних умовах	Обсяг дескриптора
SIFT	Обмежене	Так	Висока	128 float
SURF	Можливе	Так	Висока	64 float
BRIEF	Придатне	Ні	Низька	256 bit
ORB	Оптимальне	Ні	Середня	256 bit

Рисунок 1.8 – Порівняльний аналіз методів SIFT, SURF, BRIEF та ORB

Алгоритм ORB був розроблений як спроба поєднати переваги швидкодії BRIEF із адаптивністю більш складних методів. Він інтегрує покращений детектор FAST для локалізації ключових точок і модифікований дескриптор BRIEF, доповнений механізмом обчислення орієнтації. Завдяки цьому ORB забезпечує стійкість до повороту зображення при збереженні компактності та швидкодії бінарного представлення ознак. Крім того, алгоритм є повністю відкритим, не підпадає під патентні обмеження і тому активно використовується в численних системах реального часу, зокрема у вбудованих та мобільних застосунках. Водночас точність ORB дещо поступається методам SIFT і SURF в умовах складних змін масштабу, освітлення або зашумленості, що зумовлює його переважне використання у менш варіативних середовищах (рисунок 1.8).

Таким чином, кожен із розглянутих методів має свою область ефективного застосування, яка визначається компромісом між точністю,

обчислювальними витратами, стійкістю до змін і правовими обмеженнями. Вибір алгоритму залежить від конкретних вимог системи – наприклад, SIFT і SURF доцільні для високоточних задач у стабільному середовищі, тоді як ORB і BRIEF краще підходять для ресурсозалежних застосунків, орієнтованих на швидку обробку даних.

### 1.3 Методи попередньої обробки зображень

Попередня обробка зображень (рисунок 1.9) є невід’ємним етапом у будь-якому обчислювальному процесі аналізу візуальних даних, особливо якщо йдеться про застосування штучних нейронних мереж. Вона забезпечує підвищення якості вхідної інформації, зменшує вплив шумів і артефактів, а також сприяє уніфікації вхідного простору, що є критично важливим для стабільної та ефективної роботи моделей машинного навчання. Найчастіше на цьому етапі застосовуються методи фільтрації, нормалізації та масштабування, кожен із яких виконує специфічну функцію підготовки даних.

Метод	Призначення	Типові алгоритми
Фільтрація	Зменшення шумів, підсилення контурів, локальне згладжування зображення	Гауссове згладжування, медіанний фільтр, Sobel, Laplacian
Нормалізація	Уніфікація діапазону інтенсивностей пікселів для стабільного навчання нейромережі	Мін-макс нормалізація, z-нормалізація, ділення на 255
Масштабування	Приведення зображення до фіксованого розміру, сумісного з архітектурою моделі	Інтерполяція (бікв. або бікуб.), ресайз із паддінгом або crop
Метод	Переваги	Обмеження
Фільтрація	Зменшує вплив артефактів і шуму, покращує якість ознак	Може спотворити дрібні деталі при надмірному згладжуванні
Нормалізація	Забезпечує стабільність обчислень, прискорює збіжність моделі	Не враховує просторовий контекст, може втратити локальні особливості
Масштабування	Уніфікує розміри входу, знижує обчислювальну складність	Може змінити пропорції об’єкта, втратити інформацію при обрізанні

Рисунок 1.9 – Методи попередньої обробки зображень

Фільтрація зображень спрямована на усунення випадкових перешкод і локальних нерівномірностей інтенсивності пікселів, які можуть заважати подальшій обробці. В залежності від мети фільтрація може бути як низькочастотною для згладжування й усунення шуму, так і високочастотною для підсилення контурів та дрібних деталей. До найпоширеніших методів належать гауссове згладжування, медіанне фільтрування та біноміальне згортання. Особливо ефективним є застосування фільтрів до вхідних

зображень перед передачею в згорткові шари нейронних мереж, оскільки це дозволяє моделі зосередитися на семантично значущих елементах зображення.

Нормалізація інтенсивностей є ще одним важливим компонентом попередньої обробки. Вона дозволяє привести значення пікселів до єдиного масштабу, що особливо важливо для стабільності градієнтного навчання нейронної мережі. Найчастіше нормалізація здійснюється шляхом приведення значень пікселів до інтервалу  $[0, 1]$  або  $[-1, 1]$  шляхом ділення на максимальне значення (наприклад, 255 для 8-бітного зображення). Це дозволяє уникнути числових нестабільностей у процесі оптимізації, а також знижує ризик переважання окремих ділянок зображення через високі абсолютні значення інтенсивності.

Масштабування або зміна розміру зображення використовується для приведення всіх вхідних зразків до єдиного формату, який сумісний із архітектурою обраної моделі. Оскільки більшість нейронних мереж вимагають фіксованого розміру входу, масштабування дозволяє адаптувати зображення до цієї вимоги. При цьому важливо зберегти пропорції, щоб уникнути викривлення об'єктів. Залежно від контексту задачі, масштабування може також супроводжуватися обрізанням, паддінгом або інтерполяційними перетвореннями. Таке уніфікування простору входу не тільки спрощує структуру моделі, а й зменшує кількість параметрів, необхідних для її навчання.

У сукупності, методи попередньої обробки зображень спрямовані на підвищення якості даних, що подаються на вхід моделі, і водночас – на зниження впливу зовнішніх факторів, які можуть спотворити результати навчання або класифікації. Їх використання є необхідною умовою побудови ефективних систем комп'ютерного зору, що функціонують у непередбачуваних або зашумлених умовах.

## 2 ШТУЧНІ НЕЙРОННІ МЕРЕЖІ В ЗАДАЧАХ ОБРОБКИ ЗОБРАЖЕНЬ

Сучасні методи комп'ютерного зору базуються переважно на використанні штучних нейронних мереж, які продемонстрували значний прорив у якості аналізу зображень у порівнянні з традиційними алгоритмами. Основна ідея побудови ШНМ полягає в імітації принципів обробки інформації в біологічному мозку, де кожен штучний нейрон виконує просту операцію над вхідними сигналами та передає результат на наступний рівень. В архітектурному сенсі нейромережі складаються з послідовності шарів, у яких відбувається поступове витягування дедалі абстрактніших ознак, що дозволяє формувати ефективні представлення вхідних зображень навіть у зашумлених або варіативних умовах.

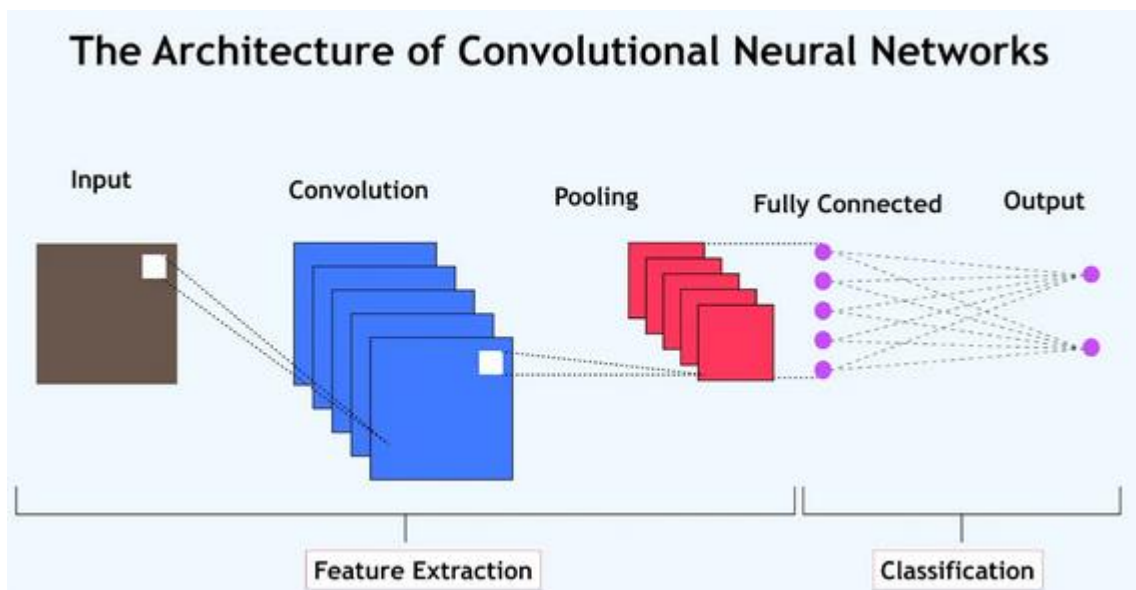


Рисунок 2.1 – Архітектура CNN

З усіх типів нейронних мереж найбільше застосування у сфері зображень отримали CNN (рисунки 2.1 та 2.2), які спеціалізуються на обробці двовимірних структур. Їхня ефективність пояснюється здатністю автоматично виявляти локальні просторові залежності за допомогою згорткових фільтрів, що переміщуються по зображенню. Початкові

архітектури, такі як LeNet, були запропоновані для задач розпізнавання рукописних цифр і стали першим успішним прикладом глибокого навчання у сфері комп'ютерного зору. Подальший розвиток цієї ідеї призвів до появи складніших моделей, таких як AlexNet, яка завдяки застосуванню GPU і ReLU-активацій вперше досягла прориву на конкурсі ImageNet. Мережі VGG поглибили цю концепцію через використання великої кількості згорткових шарів із фіксованими маленькими фільтрами, що забезпечило стабільність і простоту масштабування архітектури. ResNet запропонувала принципово новий підхід до побудови глибоких мереж шляхом введення залишкових зв'язків, які дозволяють ефективно тренувати надзвичайно глибокі структури без втрати градієнта.

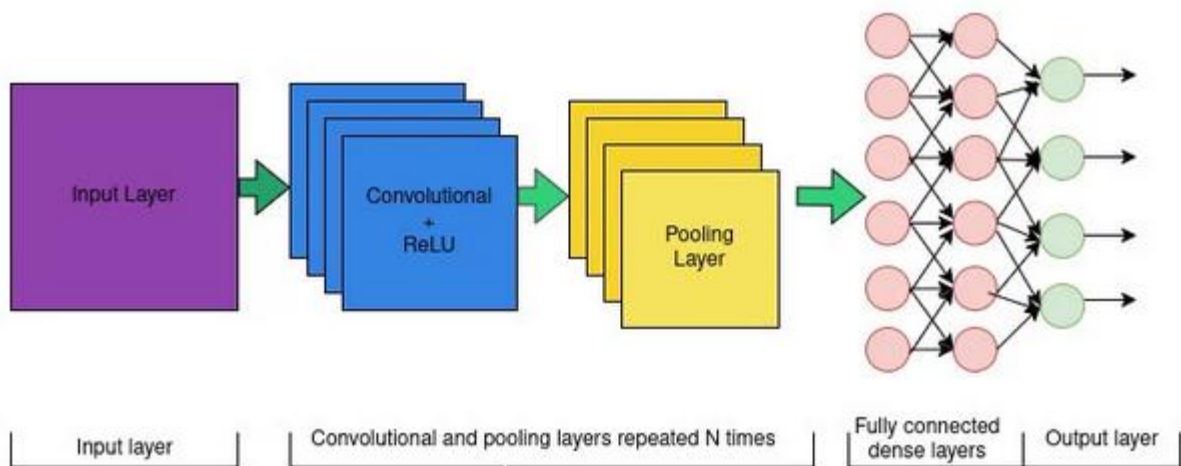


Рисунок 2.2 – Вміст CNN

Окрім класифікаційних задач, нейронні мережі широко використовуються і в більш складних задачах, таких як сегментація або відновлення зображень. У цьому контексті автоенкодери відіграють важливу роль як інструмент безнаглядного навчання. Вони навчаються стискати вхідні зображення до компактного векторного представлення з подальшим відновленням вихідного зображення з цієї латентної репрезентації. Така здатність до вивчення узагальнених ознак зображення без використання міток робить автоенкодери придатними для вирішення задач денойзингу,

генерації або пошуку аномалій. У сегментаційних задачах зазвичай використовуються симетричні архітектури, подібні до U-Net, де кожен крок згортання компенсується відповідним кроком деконволюції з поелементним поєднанням просторової інформації.

З огляду на потребу у зручній реалізації нейронних моделей, широкого поширення набули спеціалізовані бібліотеки для роботи з нейронними мережами. Однією з найпоширеніших є TensorFlow – потужна платформа для чисельних обчислень від компанії Google, яка забезпечує низькорівневий контроль над моделями. Поверх неї побудована високорівнева бібліотека Keras, яка значно спрощує процес опису, навчання і тестування нейронних мереж, роблячи його доступним навіть для користувачів без глибокого математичного бекграунду. Ще одним популярним інструментом є PyTorch – динамічний фреймворк від Facebook AI Research, що відзначається гнучкістю, простотою в дебагінгу та природною інтеграцією з Python-кодом. Обидві платформи активно розвиваються, мають потужну екосистему та підтримують запуск на GPU, що дозволяє значно прискорити процес навчання глибоких моделей [2].

Таким чином, штучні нейронні мережі, а особливо згорткові структури, автоенкодери та їхні сучасні варіації, у поєднанні з розвинутими програмними засобами реалізації, становлять потужний інструментарій для обробки зображень, що відкриває широкі можливості для досліджень і практичного застосування в багатьох галузях цифрової інженерії.

## 2.1 Основи штучних нейронних мереж

Штучні нейронні мережі є однією з фундаментальних моделей сучасного машинного навчання, яка імітує структуру та функціональні властивості біологічних нейронів. На базовому рівні нейронна мережа складається з набору пов'язаних між собою штучних нейронів, що організовані у шари. Кожен нейрон отримує на вході певний вектор значень,

виконує зважене сумування цих значень із відповідними ваговими коефіцієнтами, додає зміщення і передає результат через нелінійну активаційну функцію. Таке перетворення дозволяє моделювати складні залежності, що не описуються лінійними законами, та ієрархічно витягувати ознаки різного рівня абстракції.

Навчання нейронної мережі відбувається шляхом мінімізації функції втрат, що визначає ступінь відхилення передбачених значень від очікуваних. Для цього застосовується алгоритм зворотного поширення помилки (backpropagation), який дозволяє розрахувати похідні функції втрат за вагами на кожному шарі, після чого виконується їх оновлення за допомогою градієнтного спуску або його модифікацій. У процесі навчання мережа адаптує свої ваги таким чином, щоб покращити здатність до узагальнення, тобто до правильного передбачення результатів на нових, раніше невідомих прикладах.

Нейронні мережі можна класифікувати за глибиною, типом з'єднань та функціональним призначенням. Найпростіші – це одношарові перцептрони, здатні навчатися лише лінійно роздільним задачам. Багатшарові перцептрони (MLP) уже дозволяють апроксимувати довільні неперервні функції, але обмежені у роботі зі структурованими даними, такими як зображення. Це зумовило розвиток спеціалізованих архітектур, зокрема згорткових нейронних мереж, які використовують принцип локальних рецептивних полів і вагових спільнот, що суттєво зменшує кількість параметрів і враховує просторову структуру даних.

## 2.2 Архітектури CNN

CNN є спеціалізованим типом глибоких штучних нейронних мереж, які призначені для ефективного аналізу структурованих двовимірних даних, передусім зображень. Їхня ключова особливість полягає у використанні згорткових шарів, які імітують біологічні рецептивні поля – обмежені

ділянки простору, на які реагують окремі нейрони [3]. Завдяки цьому CNN здатні виявляти локальні шаблони, такі як контури, текстури або фрагменти об'єктів, незалежно від їх положення на зображенні. Застосування згорткових фільтрів із ваговим поділом значно зменшує кількість параметрів порівняно з повнозв'язними шарами, що робить можливим тренування глибоких моделей без надмірної складності.

Однією з перших успішних реалізацій CNN була модель LeNet-5, розроблена Янном Лекуном для розпізнавання рукописних цифр у банківських чеках. Архітектура цієї моделі складалася зі згорткових і підвибіркових шарів, за якими слідував повнозв'язний класифікатор. Незважаючи на обмеженість у глибині, LeNet-5 заклала фундаментальні принципи, що залишаються актуальними донині.

Новим етапом у розвитку згорткових мереж стала поява AlexNet (рисунок 2.4), яка здобула перемогу в престижному конкурсі ImageNet у 2012 році. Модель вирізнялася глибшою структурою, більшим обсягом даних для тренування, а також вперше активно використовувала графічні процесори (GPU) для прискорення навчання. Однією з технічних інновацій стало впровадження функції активації ReLU, яка забезпечила кращу збіжність та уникнення проблеми зникаючих градієнтів. Також було використано прийом dropout для регуляризації та шар нормалізації, що стало стандартом у подальших архітектурах.

Подальший розвиток CNN представлений архітектурою VGG, яка стала прикладом ієрархічно організованої глибокої моделі, де всі згорткові шари використовували ядра фіксованого розміру  $3 \times 3$ . Завдяки такій уніфікованій структурі, VGG стала особливо зручною для перенавчання на нових задачах та широко використовуваною в задачах transfer learning. Попри високу точність, VGG мала великий обсяг параметрів, що ускладнювало її розгортання у системах із обмеженими обчислювальними ресурсами.

Архітектура ResNet (рисунок 2.3), запропонована в 2015 році, запропонувала кардинально нову концепцію – використання залишкових

зв'язків, які дозволяють передавати вхідний сигнал напряму через кілька шарів без обчислювального перетворення. Така стратегія дозволила навчати дуже глибокі мережі, що містять понад 100 шарів, без деградації точності та втрати градієнта. ResNet відкрила шлях до створення надглибоких моделей, які досягли рекордної продуктивності на великомасштабних наборах зображень.

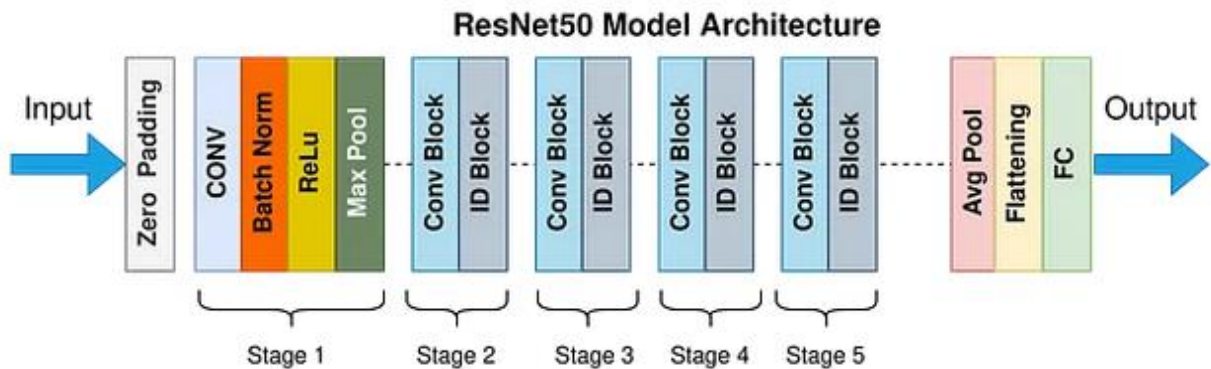


Рисунок 2.3 – Архітектура ResNet

Усі перелічені архітектури – від LeNet до ResNet – є етапами еволюції згорткових нейронних мереж, кожна з яких запропонувала новий рівень ефективності, гнучкості та глибини. Сьогодні CNN залишаються основним інструментом у розв'язанні більшості задач комп'ютерного зору, зокрема класифікації, детекції, сегментації, відстеження об'єктів та генерації зображень. Їх постійна оптимізація, розвиток і адаптація до нових обчислювальних платформ продовжують розширювати горизонти використання нейромереж у прикладних системах.

### 2.3 Автоенкодері та сегментація

У межах завдань комп'ютерного зору все більшу роль відіграють нейронні мережі, здатні працювати з неперервними, високорозмірними вхідними даними без потреби в мітках. Одним із ключових інструментів у

такому контексті стали автоенкодері – тип штучних нейронних мереж, що навчаються реконструювати вхідні дані, проходячи через вузьке латентне представлення. На відміну від класичних класифікаційних архітектур, автоенкодері реалізують симетричну структуру, що складається з енкодера, який стискає вхідний сигнал до компактного векторного коду, та декодера, що виконує зворотнє перетворення для відновлення початкового зображення. У процесі такого навчання мережа опановує здатність виявляти найінформативніші ознаки, що забезпечують ефективну репрезентацію.

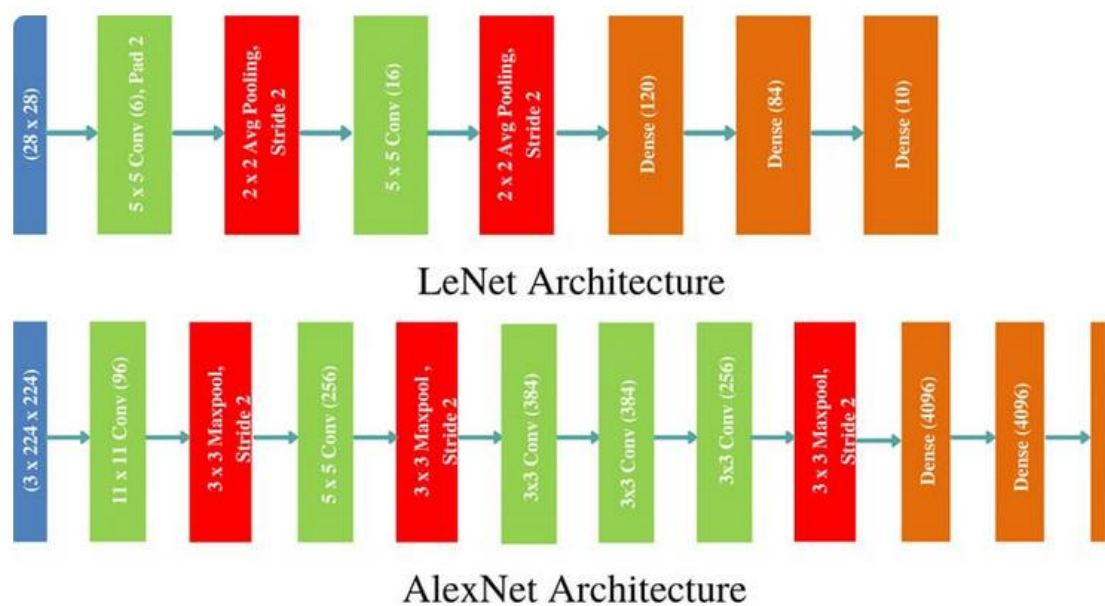


Рисунок 2.4 – LeNet та AlexNet

Особливу цінність автоенкодері мають у навчанні без вчителя, коли відсутні мітки або вони є надто дорогими для отримання. У задачах з обробки зображень автоенкодері застосовуються для зменшення розмірності, усунення шуму, відновлення зіпсованих фрагментів, виявлення аномалій або стиску зображень. Їхні модифікації, такі як денойзингові автоенкодері, навчаються на зашумлених вхідних даних і демонструють здатність до відновлення чистих версій без потреби в ручному втручанні. Варіаційні автоенкодері (VAE), своєю чергою, поєднують навчання представлення з елементами ймовірнісного моделювання, дозволяючи

формувати згладжені латентні простори, зручні для генеративних задач.

Водночас важливим напрямом застосування нейромереж у візуальних даних є семантична сегментація, що передбачає класифікацію кожного пікселя зображення за певною категорією. У цьому контексті ефективними виявились симетричні архітектури, які поєднують принципи згорткових мереж і автоенкодерів. Типовий приклад – архітектура U-Net, побудована на основі принципу «стиснення–розгортання», де енкодер витягує ієрархію ознак, а декодер відновлює просторову структуру з урахуванням отриманих представлень. Ключовою перевагою таких моделей є наявність з'єднань між відповідними рівнями енкодера і декодера (skip-connections), що забезпечує точну локалізацію об'єктів навіть у глибоких мережах[5].

Сегментаційні нейромережі використовуються в різноманітних галузях – від медичної діагностики, де необхідно ізолювати патологічні області на знімках, до автоматичного розпізнавання дорожніх сцен в автономному транспорті. Завдяки здатності враховувати як контекст, так і локальні деталі, такі архітектури забезпечують високу точність і надійність при обробці складних зображень [6].

Таким чином, автоенкодери і сегментаційні мережі відіграють ключову роль у сучасному арсеналі глибокого навчання для зображень. Вони не лише дозволяють розв'язувати задачі генерації та реконструкції, а й формують основу для точного, контекстно-залежного аналізу візуальних даних, розширюючи можливості застосування нейромереж у сферах з обмеженими ресурсами або недостатнім маркуванням.

## 2.4 Бібліотеки реалізації нейронних мереж

Успішне впровадження моделей глибокого навчання, зокрема штучних нейронних мереж, стало можливим завдяки активному розвитку програмних бібліотек, які забезпечують високопродуктивні обчислення, гнучке моделювання та інтерфейси для розробки й навчання моделей. З-поміж

численних інструментів, найбільшу популярність у сфері комп'ютерного зору та обробки зображень здобули TensorFlow, Keras і PyTorch, які стали стандартом для досліджень, прототипування й індустріального впровадження.

TensorFlow, розроблений компанією Google, є однією з найпотужніших платформ для реалізації обчислень із підтримкою графічних процесорів (GPU) та розподілених середовищ. Він реалізує обчислення у вигляді графа залежностей, що дозволяє ефективно керувати пам'яттю, кешувати проміжні обчислення та забезпечувати гнучку оптимізацію. TensorFlow має багатий набір функціональних API для створення нейронних мереж різної складності, а також підтримку експорту моделей у форматі TensorFlow Lite для використання на мобільних та вбудованих пристроях. Одним із ключових переваг цієї платформи є її інтеграція з інструментами для візуалізації, хмарного навчання і автоматизованого підбору архітектур [4].

Паралельно з TensorFlow активно розвивається Keras – високорівнева бібліотека, яка зосереджена на простоті опису моделей, швидкому прототипуванні та читабельності коду. Вона забезпечує доступ до базових компонентів нейронної мережі – шарів, функцій втрат, оптимізаторів і регуляризаторів – через лаконічний інтерфейс, що зменшує поріг входу для новачків. Початково Keras існувала як окрема бібліотека з можливістю підключення різних бекендів, але з часом стала офіційним високорівневим API TensorFlow. Завдяки Keras процес побудови згорткових мереж, автоенкодерів або рекурентних архітектур спрощується до кількох десятків рядків коду, при цьому зберігається доступ до всіх можливостей TensorFlow для масштабування, моніторингу та розгортання моделей.

Іншою популярною альтернативою є PyTorch, створений дослідницьким підрозділом компанії Meta. Основною відмінністю цієї бібліотеки є динамічна побудова обчислювального графа, яка надає більшу гнучкість під час відлагодження та дає змогу змінювати архітектуру моделі "на льоту". PyTorch особливо популярний серед дослідників завдяки

прозорості механізмів виконання, зручній роботі з тензорами, прямій інтеграції з Python та активній спільноті. Наявність бібліотек надбудови, таких як Torchvision, Torchtext, FastAI, розширює функціональність базового ядра, забезпечуючи доступ до готових моделей, передоброби даних та інструментів для глибокої оптимізації.

Попри певні відмінності у внутрішній реалізації, усі ці фреймворки надають повний інструментарій для реалізації, навчання та валідації нейронних мереж, включаючи засоби моніторингу, збереження моделей, використання GPU/TPU, підтримку автоматичного диференціювання та інтеграцію з хмарними обчислювальними середовищами. Усі три бібліотеки також підтримуються на платформі Google Colab, що дозволяє без додаткових локальних ресурсів реалізовувати і тестувати повноцінні проекти з обробки зображень на основі глибокого навчання.

Таким чином, вибір бібліотеки реалізації залежить від конкретних вимог проекту: TensorFlow підходить для масштабованих систем і промислового розгортання, Keras – для швидкого прототипування та освітніх цілей, а PyTorch – для гнучких дослідницьких завдань і відлагодження нових архітектур. Разом вони створюють екосистему інструментів, що значно спрощує роботу з нейронними мережами та прискорює інновації у сфері комп'ютерного зору.

## 2.5 Обґрунтування вибору ШНМ для поставленого завдання

У контексті завдання обробки зображень доцільно використовувати CNN [7], які спеціалізовані на обробці двовимірних масивів пікселів. Основні переваги CNN перед традиційними багатошаровими MLP або RNN полягають у їхній здатності:

- автоматично виділяти локальні ознаки (edges, textures, shapes) завдяки операціям згортки;
- мати менше параметрів завдяки розділенню ваг фільтрів, що

зменшує обчислювальну складність;

- зберігати просторову структуру зображення, на відміну від MLP, який перетворює вхід на плоский вектор;

- забезпечувати високу точність при класифікації зображень, що підтверджено результатами численних досліджень (рисунок 2.5).

Архітектура	Призначення	Переваги	Недоліки
MLP	Загальні задачі класифікації	Простота реалізації	Не враховує структуру зображення, багато параметрів
CNN	Аналіз зображень	Локальні фільтри, просторові зв'язки, економія параметрів	Може потребувати більше ресурсів
RNN / LSTM	Обробка послідовностей (текст, мова)	Модель пам'яті, працює з часовими рядами	Не підходить для просторових зображень
Autoencoder	Зниження розмірності, денойзінг	Компресія зображень, навчання без міток	Не оптимальна для класифікації
Transformers (ViT)	Обробка зображень (візуальні трансформери)	Потужні для великих наборів	Вимагають багато даних і ресурсів

Рисунок 2.5 – Порівняльна таблиця архітектур

Згорткові нейронні мережі є найефективнішим вибором для задачі обробки зображень завдяки здатності виявляти багаторівневі абстракції (від країв до форм і об'єктів), економному використанню пам'яті та підтвердженій ефективності у практичних задачах. У межах цієї кваліфікаційної роботи CNN було обрано як базову архітектуру для трьох задач класифікації зображень на наборах даних MNIST, Fashion-MNIST та CIFAR-10.

## 3 РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ

### 3.1 Вибір інструментів і середовища розробки

У процесі розробки програмного засобу для обробки зображень з використанням штучних нейронних мереж було обрано інструменти, які поєднують потужні функціональні можливості, широку підтримку спільноти та зручність розгортання у хмарному середовищі. Основною платформою для реалізації обчислювальних експериментів стало інтерактивне середовище Google Colab, що надає користувачеві змогу запускати Jupyter-ноутбуки без потреби у локальній конфігурації апаратного забезпечення. Однією з ключових переваг цієї платформи є підтримка апаратного прискорення (GPU, TPU), що дозволяє значно пришвидшити процес навчання глибоких нейронних мереж, зменшуючи час експериментального циклу.

Google Colab інтегрується з хмарним сховищем Google Drive, що забезпечує зручність у збереженні даних, моделей та результатів візуалізації. Можливість одночасної роботи над проектом у спільному доступі також відкриває перспективи для командної співпраці, що є особливо актуальним у контексті навчальних, дослідницьких або прикладних проектів.

Для реалізації моделей нейронних мереж у проекті було використано TensorFlow – потужний фреймворк від компанії Google, що поєднує гнучкість побудови обчислювальних графів і широкі можливості для оптимізації. Його застосування забезпечує повний контроль над процесом навчання, включно з налаштуванням функцій втрат, оптимізаторів, параметрів регуляризації та моніторингу збіжності моделі. Поверх TensorFlow використано Keras – високорівневу бібліотеку, яка значно спрощує опис архітектури моделі та її навчання, дозволяючи зосередитися на логіці дослідження, а не на низькорівневих деталях реалізації.

Для підготовки даних, їх візуалізації та статистичного аналізу

використано бібліотеки NumPy, Matplotlib та Seaborn. NumPy забезпечує ефективну роботу з багатовимірними масивами даних, що є основним типом структури зображень у процесі обробки. Matplotlib і Seaborn дозволяють будувати графіки точності, втрат, матриці плутанини та інші діаграми, що є необхідними для інтерпретації роботи моделі й представлення результатів.

Загалом, обрані інструменти створюють цілісну екосистему, яка охоплює всі етапи життєвого циклу моделі – від завантаження даних і попередньої обробки, до побудови архітектури, навчання, валідації та оцінювання результатів. Завдяки використанню відкритих технологій та хмарної інфраструктури реалізація залишається доступною, масштабованою та адаптованою для подальших модифікацій чи перенесення в інші середовища (рисунок 3.1).



Рисунок 3.1 – Вибір програмних засобів

### 3.2 Обґрунтування вибору датасетів

Для повноцінного аналізу ефективності штучних нейронних мереж у задачах обробки та класифікації зображень у цій роботі було обрано три найбільш репрезентативні відкриті набори даних: MNIST, Fashion-MNIST та

CIFAR-10. Кожен із цих датасетів має різну природу, складність, глибину представлення ознак, що дозволяє протестувати моделі на широкому спектрі вхідних образів – від простих рукописних цифр до кольорових реалістичних об'єктів (рисунок 3.2).

Набір даних MNIST є класичним еталоном для задач класифікації рукописних цифр. Він містить 60 000 зображень для навчання та 10 000 для тестування, кожне з яких має розмір  $28 \times 28$  пікселів у відтінках сірого. Простота структури зображень у поєднанні з чіткою класифікаційною постановкою задачі дозволяє використовувати MNIST як базовий контрольний приклад для оцінки базових характеристик моделей. Він є особливо корисним для початкової перевірки архітектури мережі, налаштування гіперпараметрів і візуального аналізу навчання.

Датасет Fashion-MNIST є вдосконаленим аналогом MNIST, що містить зображення предметів одягу та взуття з класифікацією на 10 категорій (футболки, штани, кросівки тощо). Кожне зображення також має розмір  $28 \times 28$  пікселів у відтінках сірого. Попри схожу структуру, цей набір даних є значно складнішим з точки зору міжкласової подібності та візуальної неоднозначності, що дає змогу перевірити здатність мережі до навчання на більш тонких відмінностях ознак.

На відміну від двох попередніх, CIFAR-10 є набором кольорових зображень розміром  $32 \times 32$  пікселі, що охоплюють 10 різних категорій реальних об'єктів, зокрема літаки, автомобілі, птахів, котів, собак, кораблі тощо. Набір містить 50 000 зображень для навчання і 10 000 для тестування. Його використання дозволяє перевірити ефективність згорткових нейронних мереж у контексті роботи з тривимірними (RGB) образами, багатими на текстури, колір і шумові артефакти. CIFAR-10 широко застосовується в дослідницьких експериментах і слугує перевіреним інструментом порівняння моделей різної складності.

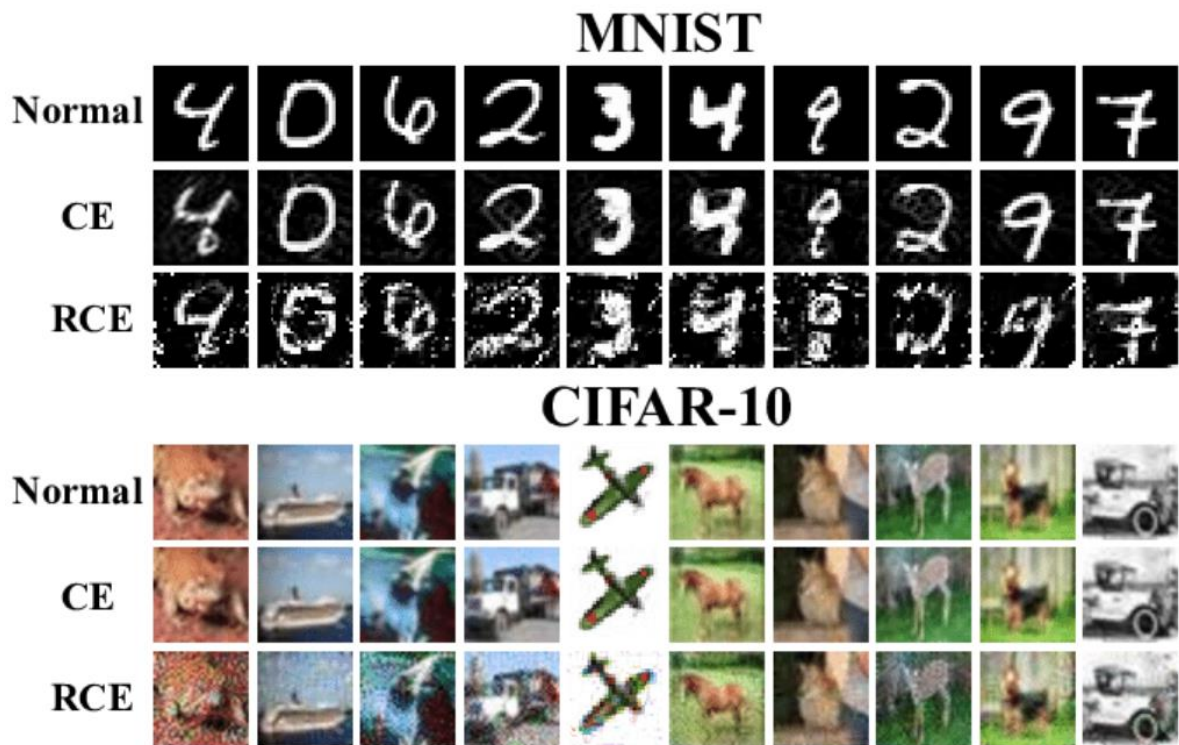


Рисунок 3.2 – Набори даних

Використання трьох зазначених датасетів дозволяє не лише оцінити адаптивність архітектури до зображень з різним ступенем складності, але й дослідити здатність мережі до узагальнення, стабільність навчального процесу та якість класифікації. Такий підхід забезпечує системний підхід до тестування, уможливорюючи формування обґрунтованих висновків про переваги та обмеження вибраної архітектури нейронної мережі.

### 3.3 Реалізація та аналіз результатів

Крок 1. Імпорт необхідних бібліотек:

- tensorflow та keras: основні інструменти для побудови та навчання нейронної мережі;
- matplotlib.pyplot, seaborn: для візуалізації зображень та результатів;
- sklearn.metrics: для побудови та виводу матриці плутанини.

Крок 2. Завантаження та попередня обробка даних:

- завантажується MNIST – набір зображень рукописних цифр;
- дані нормалізуються до діапазону  $[0, 1]$ .
- додається вимір для каналу (зображення перетворюються на  $28 \times 28 \times 1$ ), що потрібно для згорткової мережі.

Крок 3. Візуалізація прикладів з навчального набору:

- виводяться перші 9 зображень з підписами класів (рисунок 3.3).
- забезпечується первинний візуальний контроль якості даних після попередньої обробки.

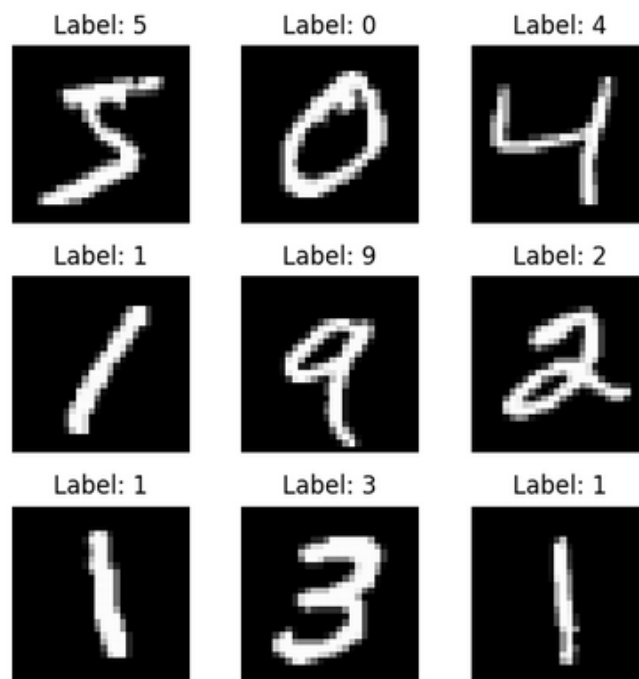


Рисунок 3.3 – Вивід перших 9 зображень

На наступному етапі було сформовано архітектуру згорткової нейронної мережі для класифікації зображень із датасету MNIST. Було використано багаторівневу модель з двома згортковими шарами та шарами підвибірки, що дозволяє поступово зменшувати розмірність вхідного зображення, зберігаючи ключові ознаки. Після згорткових операцій, виконувалася операція «плоского» представлення ознак у вектор для подальшої передачі до повнозв'язного шару. Завершальний шар мережі мав

10 виходів, що відповідають класам цифр. Для функції втрат обрано крос-ентропію, а оптимізацію здійснював алгоритм Adam. Було згенеровано зведення моделі, що дозволило оцінити кількість параметрів і глибину мережі.

Після побудови моделі (рисунок 3.4) розпочався процес навчання, який проводився протягом п'яти епох з використанням як навчальної, так і тестової вибірки. В ході навчання мережі фіксувалася точність класифікації, що надалі виводилася графічно для оцінювання динаміки збіжності моделі.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (Dense)	(None, 64)	102,464
dense_1 (Dense)	(None, 10)	650

Рисунок 3.4 – Побудова CNN для MNIST

У продовження роботи було підключено другий набір даних – Fashion-MNIST. Його обробка здійснювалася аналогічно: попереднє масштабування зображень до діапазону  $[0, 1]$ , а також додавання каналу до зображень. Для попередньої візуальної оцінки якості та змісту зображень було побудовано сітку з 9 прикладів, кожен з яких мав відповідну мітку.

Для задачі класифікації зображень одягу було сконструйовано окрему згорткову нейронну мережу, архітектурно ідентичну тій, яка застосовувалася до MNIST. Модель було скомпільовано з однаковими параметрами, після чого здійснювалося навчання протягом п'яти епох із подальшою перевіркою на тестовій вибірці. Така уніфікованість підходу дозволяє порівнювати результати на різних наборах даних при сталих архітектурних і навчальних умовах, що забезпечує об'єктивність аналізу.

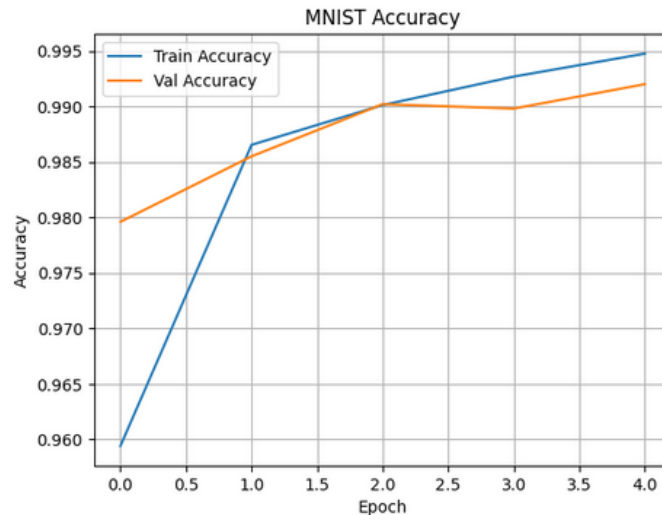


Рисунок 3.5 – Візуалізація процесу навчання

Графік, представлений на рисунку 3.5, ілюструє зміну точності класифікації (accuracy) моделі згорткової нейронної мережі під час навчання на датасеті MNIST упродовж п'яти епох. По осі абсцис відкладається номер епохи, тоді як по осі ординат – значення точності на тренувальній та валідаційній вибірках. На графіку порівнюються дві криві: Train Accuracy (синя лінія) та Val Accuracy (помаранчева лінія), які відповідно відображають ефективність моделі на навчальному та тестовому підмножинах даних.

Уже на першій епісі спостерігається високий рівень точності, що свідчить про здатність мережі швидко адаптуватися до структури вхідних даних. Подальше тренування демонструє стабільне зростання точності на обох вибірках, при цьому приріст є плавним, без різких коливань або спадів. Важливо зазначити, що валідаційна точність не лише не відстає від тренувальної, але в окремі моменти навіть її перевищує, що може свідчити про добре підбрану архітектуру, ефективну регуляризацию або оптимальні гіперпараметри.

Форма кривих свідчить про збіжність моделі та відсутність ознак перенавчання у рамках досліджуваного інтервалу. Збіжність обох кривих у правій частині графіка свідчить про узгодженість між фазами навчання та тестування. Це підтверджує, що побудована нейронна мережа має здатність

до узагальнення й демонструє високу якість класифікації рукописних цифр навіть на невідомих даних.

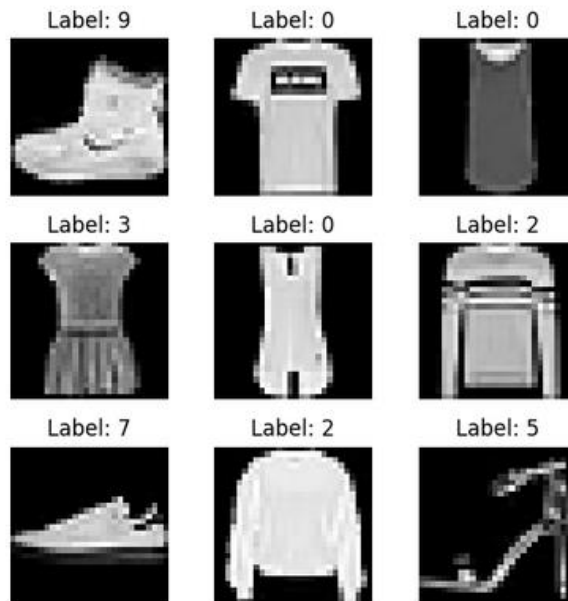


Рисунок 3.6 – Візуалізація прикладів Fashion-MNIST

Зображення ілюструє фрагмент навчального набору Fashion-MNIST, що є уніфікованим набором зображень предметів одягу для задач класифікації. На рисунку наведено 9 сіро-білих зображень розміром  $28 \times 28$  пікселів, кожне з яких супроводжується відповідною міткою (Label), що позначає клас, до якого належить об'єкт. Серед зображень представлено різноманітні типи одягу та аксесуарів, зокрема кросівки (Label: 9), футболки (Label: 0), сукні (Label: 3, 7), светри (Label: 2), туфлі на підборах (Label: 5) та сумки (Label: 2).

Зображення відображають характерну візуальну специфіку датасету Fashion-MNIST: низька роздільна здатність, сіро-біле кодування та контрастність між об'єктом і фоном (рисунок 3.6). Незважаючи на обмеженість просторової інформації, елементи предметів чітко проглядаються, що робить набір придатним для навчання згорткових нейронних мереж.

Вибрані приклади демонструють варіативність об'єктів одного класу

(наприклад, футболки Label: 0) та відносну візуальну близькість між деякими категоріями (наприклад, між сукнею і тунікою), що зумовлює складність задачі класифікації. Така візуалізація є важливим етапом перед початком навчання моделі, оскільки дозволяє переконатися у якості завантажених даних, їх відповідності очікуваним класам (рисунок 3.7), а також оцінити потенційні труднощі для нейромережі при розрізненні схожих типів об'єктів.

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_2 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_3 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_3 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten_1 (Flatten)	(None, 1600)	0
dense_2 (Dense)	(None, 64)	102,464
dense_3 (Dense)	(None, 10)	650

Рисунок 3.7 – Побудова CNN для Fashion-MNIST

Графік, представлений на рисунку 3.8, відображає динаміку точності класифікації під час навчання згорткової нейронної мережі на основі датасету Fashion-MNIST. По осі абсцис відкладено номер епохи, тоді як по осі ординат – значення точності на тренувальній та валідаційній вибірках. Дві криві – синя для навчального набору та помаранчева для валідаційного – дозволяють візуально порівняти ефективність моделі на етапі тренування та тестування.

Вже з першої епохи спостерігається високий рівень точності, що свідчить про швидку здатність мережі адаптуватися до структурованого зображення одягу. Подальші етапи навчання демонструють поступове зростання точності, причому тренувальна вибірка показує стабільну й монотонну динаміку. Точність на валідаційній вибірці зростає повільніше, але також демонструє позитивну тенденцію, що свідчить про відсутність значного перенавчання протягом перших п'яти епох.

Хоча між кривими спостерігається певна розбіжність, вона залишається в межах прийнятної норми й не свідчить про втрату здатності до узагальнення. Рівень точності, досягнутий на обох множинах, перевищує 90% до завершення п'ятої епохи, що є хорошим показником для складнішого за MNIST набору, який містить значну міжкласову візуальну подібність.

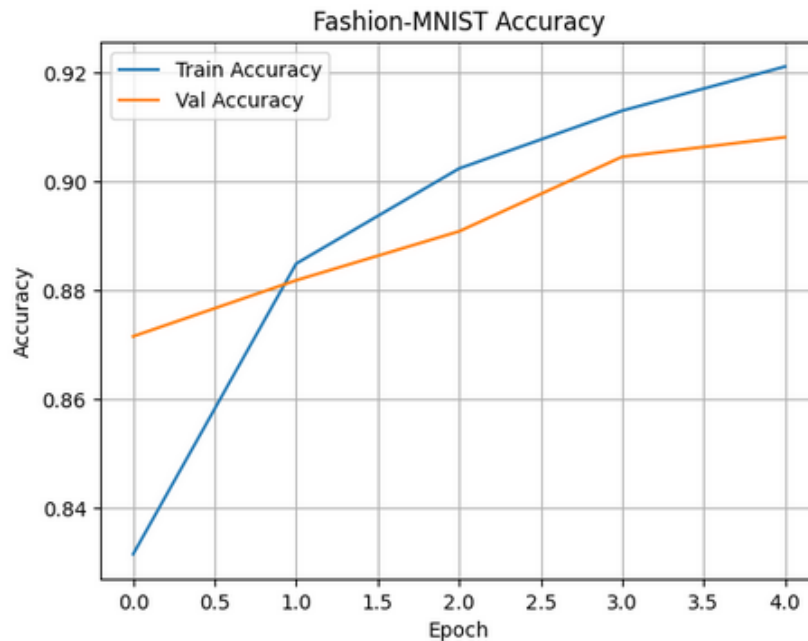


Рисунок 3.8 – Візуалізація процесу навчання Fashion-MNIST

На рисунку 3.9 представлено набір прикладів із навчальної множини датасету CIFAR-10, який є одним із найбільш вживаних еталонів у задачах глибокого навчання на кольорових зображеннях. Кожне зображення має фіксований розмір  $32 \times 32$  пікселі та складається з трьох кольорових каналів (RGB). У візуалізованій сітці зображено 9 об'єктів, кожен з яких супроводжується цифрою міткою класу (Label), що відповідає одній із 10 категорій.

Представлені приклади охоплюють різноманітні класи, зокрема об'єкти тваринного світу (кінь, олень, жаба, птах), транспортні засоби (вантажівка, легковий автомобіль, корабель), а також сцени з обладнанням або технікою (трактор, екскаватор). Класи, зашифровані у вигляді числових міток, відповідають таким категоріям, як: 0 – літак, 1 – автомобіль, 2 – птах, 3 –

кішка, 4 – олень, 5 – собака, 6 – жаба, 7 – кінь, 8 – корабель, 9 – вантажівка. Візуальне розмаїття та високий ступінь складності зображень робить CIFAR-10 більш реалістичним і викликає додаткові труднощі при класифікації, особливо враховуючи невелику роздільну здатність і подібність між окремими класами.



Рисунок 3.9 – Візуалізація прикладів CIFAR-10

Зображення демонструють як внутрішньокласову варіативність, так і міжкласову візуальну близькість, що посилює складність задачі класифікації. Наприклад, об'єкти класу "автомобіль" (Label 1) можуть істотно відрізнятися за кольором, орієнтацією та фоном, що вимагає від моделі здатності до генералізації високого рівня.

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d_4 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_5 (Conv2D)	(None, 13, 13, 64)	18,496
max_pooling2d_5 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_6 (Conv2D)	(None, 4, 4, 64)	36,928
flatten_2 (Flatten)	(None, 1024)	0
dense_4 (Dense)	(None, 64)	65,600
dense_5 (Dense)	(None, 10)	650

Рисунок 3.10 – Побудова CNN для CIFAR10

Графік, що зображено на рисунку 3.11, демонструє процес навчання згорткової нейронної мережі на наборі даних CIFAR-10 (рисунок 3.10), який характеризується вищою складністю порівняно з MNIST та Fashion-MNIST через більшу роздільну здатність зображень, наявність кольору та ширшу категоріальну варіативність. По осі абсцис відкладено кількість епох, а по осі ординат – точність класифікації, що вимірюється окремо на тренувальній та валідаційній вибірках.

У ході перших етапів навчання спостерігається поступове зростання точності як на тренувальних, так і на валідаційних даних. Початкова точність на тренувальному наборі перебуває на рівні близько 45%, тоді як валідаційна точність стартує з вищого показника – приблизно 55%. Цей ефект може бути зумовлений особливостями ініціалізації ваг або характеристиками мінібатчів у перших ітераціях.

Подальші етапи навчання характеризуються стабільним покращенням обох метрик. До завершення п'ятої епохи тренувальна точність перевищує 70%, тоді як валідаційна стабілізується на рівні близько 68%. Відносно невеликий розрив між обома кривими свідчить про відсутність суттєвого перенавчання, що є ознакою правильно підібраної архітектури мережі, регуляризаційних засобів і гіперпараметрів. Наявність поступового збільшення точності вказує на потенціал моделі для подальшого удосконалення в разі продовження тренування або модифікації структури мережі.

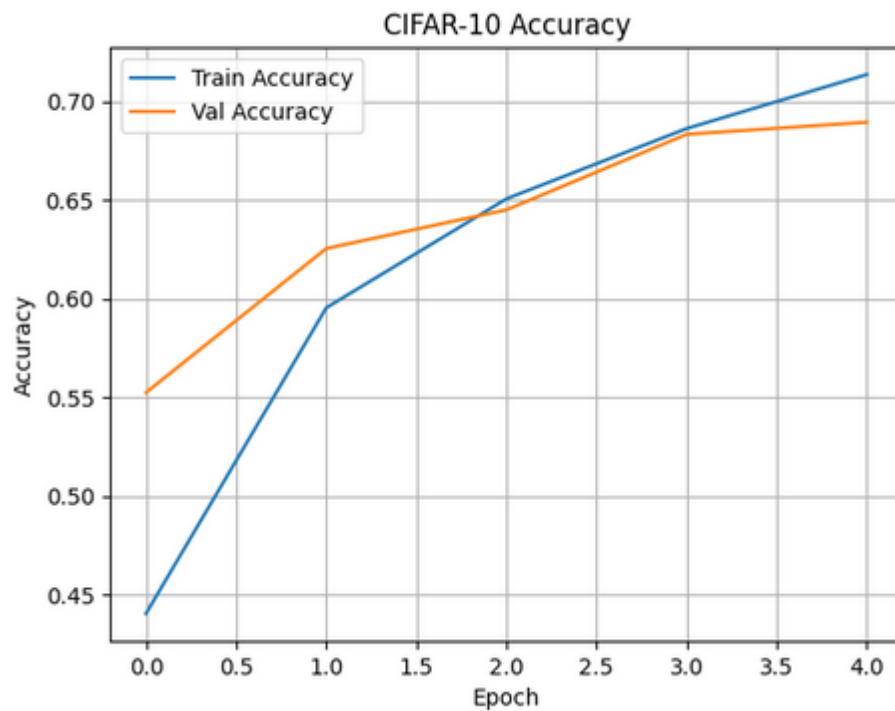


Рисунок 3.11 – Візуалізація процесу навчання CIFAR-10

Порівняно з графіками, отриманими для MNIST і Fashion-MNIST, ця крива ілюструє складніші умови навчання, що цілком очікувано для датасету CIFAR-10. Успішне наближення точності до 70% уже на п'ятій епосі свідчить про ефективність обраної конфігурації мережі навіть без глибокої оптимізації.

## ВИСНОВКИ

У результаті проведеного дослідження було реалізовано програмний засіб для обробки та класифікації зображень з використанням штучних нейронних мереж у середовищі Google Colab. У роботі було поєднано теоретичний аналіз основ глибокого навчання з практичною реалізацією згорткових нейронних мереж, що дозволило системно дослідити ефективність застосування CNN-архітектур для розпізнавання образів на прикладі трьох репрезентативних датасетів: MNIST, Fashion-MNIST та CIFAR-10.

Проведено обґрунтований вибір кожного з наборів даних: MNIST слугував для перевірки базової архітектури, Fashion-MNIST – для оцінки здатності моделі до класифікації об'єктів зі складнішою внутрішньокласовою варіативністю, а CIFAR-10 – як більш складний реалістичний набір зображень, що містить колірні канали та широку предметну категоризацію. Для кожного з наборів було здійснено відповідну попередню обробку, зокрема нормалізацію та приведення розмірності, що забезпечило коректне функціонування моделей.

Архітектура згорткової нейронної мережі була побудована з урахуванням ключових принципів CNN: згорткові шари для автоматичного витягування ознак, шари підвибірки для зменшення розмірності та повнозв'язні шари для класифікації. Навчання проводилось із застосуванням оптимізатора Adam і функції втрат категоріальної крос-ентропії, що дозволило досягти швидкої збіжності моделей. Усі реалізації були протестовані на окремих тестових вибірках, а результати оцінено за допомогою метрик точності, графіків тренування та матриць плутанини.

За результатами експериментів було встановлено, що модель досягає високої точності на простих наборах (MNIST – понад 99%, Fashion-MNIST – понад 91%), тоді як на складнішому CIFAR-10 – на рівні 70% після п'яти

epoch навчання. Отримані результати підтверджують як здатність згорткових мереж до генералізації, так і їх обмеження в умовах високої варіативності вхідних зображень. Водночас спостереження за динамікою навчання свідчать про відсутність перенавчання, що вказує на правильну обрану глибину мережі та параметри навчального процесу.

Загалом, робота підтвердила ефективність використання згорткових нейронних мереж у задачах класифікації зображень різної складності. Результати можуть бути використані як основа для подальших досліджень у напрямі оптимізації архітектур, автоматичного підбору гіперпараметрів, а також розширення до задач семантичної сегментації, локалізації об'єктів або генеративного моделювання візуального контенту.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hande Alemdar, Vincent Leroy, Adrien Prost-Boucle, and Fr'ed'eric P'etrot. Ternary neural networks for resource-efficient AI applications. In International Joint Conference on Neural Networks, 2017.
2. Genevera Allen. Sparse higher-order principal components analysis. In International Conference on Artificial Intelligence and Statistics (AISTATS), 2012.
3. Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Fixed point optimization of depth convolutional neural networks for object recognition. In Acoustics, Speech, and Signal Processing (ICASSP), International Conference on, 2015.
4. M. Astrid and Seung-Ik Lee. Cp-decomposition with tensor power method for convolutional neural networks compression. In Big Data and Smart Computing, 2017.
5. Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
6. Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In International Conference on Computer Vision (ICCV), 2015.
7. Hessam Bagherinezhad, Mohammad Rastegari, and Ali Farhadi. LCNN: Lookup-based convolutional neural network. Conference on Computer Vision and Pattern Recognition (CVPR), 2017.