

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)

Комп'ютерна система обробки зображень з  
використанням Deep Learning

(тема)

Виконав:

здобувач 4 року навчання,

групи КІУКІ-21-2

Артем КАРНОЖИЦЬКИЙ

(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: ст. викл. Андрій БУГРІЙ

(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 \_\_\_\_\_ Карножицькому Артему Олеговичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система обробки зображень з використанням Deep Learning

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 \_\_\_\_\_

глибоке навчання \_\_\_\_\_

комп'ютерна система \_\_\_\_\_

аналіз існуючих систем \_\_\_\_\_

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

Теоретичні основи обробки зображень та машинного навчання \_\_\_\_\_

Аналіз сучасних методів обробки зображень на основі машинного навчання \_\_\_\_\_

Реалізація системи обробки зображень \_\_\_\_\_

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 архітектури системи	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)

ст. викл. Андрій БУГРІЙ

(посада, власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 48 с., 11 рис., 2 дод., 7 джерел.

ГЛИБИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, TENSORFLOW, GOOGLE COLAB, МАШИННЕ НАВЧАННЯ, ОБРОБКА ЗОБРАЖЕНЬ, МОДЕЛЬ ШТУЧНОГО ІНТЕЛЕКТУ, АНАЛІЗ ДАНИХ.

Метою кваліфікаційної роботи є розробка комп'ютерної системи обробки зображень з використанням Deep Learning.

У процесі виконання кваліфікаційної роботи було розроблено та реалізовано комп'ютерну систему обробки зображень із використанням технологій глибокого навчання, зокрема згорткових нейронних мереж. Робота охоплює повний цикл побудови моделі – від аналізу предметної області й постановки задачі до реалізації, навчання, тестування та візуалізації результатів у середовищі Google Colab.

## ABSTRACT

Bachelor's thesis: 48 pages, 11 figures, 2 appendices, 7 sources.

DEEP LEARNING, COMPUTER VISION, IMAGE CLASSIFICATION, CONVOLUTIONAL NEURAL NETWORK, TENSORFLOW, GOOGLE COLAB, MACHINE LEARNING, IMAGE PROCESSING, ARTIFICIAL INTELLIGENCE MODEL, DATA ANALYSIS.

The major goal of this thesis is to develop a computer vision system for image processing using deep learning techniques.

In order to a computer system for image processing was designed and implemented using deep learning technologies, in particular convolutional neural networks. The work covers the full model development cycle — from domain analysis and problem formulation to implementation, training, testing, and visualization of results in the Google Colab environment.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	7
ВСТУП .....	8
1 ТЕОРЕТИЧНІ ОСНОВИ ОБРОБКИ ЗОБРАЖЕНЬ ТА МАШИННОГО НАВЧАННЯ .....	9
1.1 Основні поняття та визначення обробки зображень .....	9
1.2 Методи класичної обробки зображень .....	10
1.3 Поняття і принципи роботи машинного навчання .....	11
1.4 Класифікація алгоритмів машинного навчання.....	13
1.5 Використання нейронних мереж для обробки зображень.....	14
2 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ОБРОБКИ ЗОБРАЖЕНЬ НА ОСНОВІ МАШИННОГО НАВЧАННЯ.....	16
2.1 Огляд популярних нейронних архітектур .....	16
2.2 Методи підготовки та попередньої обробки даних .....	17
2.3 Вибір та оптимізація гіперпараметрів моделей .....	19
2.4 Програмні засоби і бібліотеки для роботи з зображеннями .....	20
3 РЕАЛІЗАЦІЯ СИСТЕМИ ОБРОБКИ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ.....	23
3.1 Процес розробки КС .....	24
3.2 Архітектура системи .....	25
3.3 Вибір моделі .....	26
3.4 Обробка та підготовка даних .....	27
3.5 Реалізація в Google Colab .....	28
ВИСНОВКИ.....	36
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	37
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	38
ДОДАТОК Б Програмний код.....	46
Б.1 Лістинг коду .....	46

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- API – програмний інтерфейс застосунків
- CNN – згортоква нейронна мережа (Convolutional Neural Network)
- CPU – центральний процесор (Central Processing Unit)
- CV – комп’ютерний зір (Computer Vision)
- DL – глибинне навчання (Deep Learning)
- F1-score – середньоарифметичне точності та повноти (F-measure)
- GPU – графічний процесор (Graphics Processing Unit)
- KPI – ключові показники ефективності (Key Performance Indicators)
- ML – машинне навчання (Machine Learning)
- MNIST – Modified National Institute of Standards and Technology
- NLP – обробка природної мови (Natural Language Processing)
- ReLU – випрямлена лінійна одиниця (Rectified Linear Unit)
- ResNet – залишкова мережа (Residual Network)
- TPU – тензорний процесорний блок (Tensor Processing Unit)
- U-Net – мережа U-подібної форми для сегментації зображень
- ROC – крива робочих характеристик приймача (Receiver Operating Characteristic)
- TP – істинно позитивні (True Positives)
- TN – істинно негативні (True Negatives)
- FP – хибно позитивні (False Positives)
- FN – хибно негативні (False Negatives)

## ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімким зростанням обсягів візуальної інформації та потребою в її автоматизованій обробці. Традиційні методи аналізу зображень, засновані на класичних алгоритмах комп'ютерного зору, все частіше поступаються місцем інноваційним підходам, що базуються на принципах машинного навчання та штучного інтелекту. Цей перехід обумовлений не лише технологічним прогресом, але й фундаментальними змінами у розумінні природи візуальної інформації та способів її інтерпретації.

Актуальність дослідження визначається зростаючою роллю автоматизованої обробки зображень у різноманітних галузях людської діяльності, від медичної діагностики до промислового контролю якості, від систем безпеки до творчих індустрій. Впровадження методів машинного навчання в цю сферу відкриває нові можливості для розв'язання складних завдань класифікації, сегментації, розпізнавання та генерації візуального контенту.

Мета роботи – розробка комп'ютерної системи обробки зображень з використанням Deep Learning. Для досягнення поставленої мети визначено наступні завдання: здійснення аналізу теоретичних основ обробки зображень та машинного навчання, дослідження сучасних методів та архітектур нейронних мереж, практична реалізація алгоритму обробки зображень, експериментальна перевірка ефективності розробленого методу та оцінка перспектив розвитку даного напрямку досліджень.

# 1 ТЕОРЕТИЧНІ ОСНОВИ ОБРОБКИ ЗОБРАЖЕНЬ ТА МАШИННОГО НАВЧАННЯ

## 1.1 Основні поняття та визначення обробки зображень

Обробка зображень як наукова дисципліна являє собою міждисциплінарну галузь, що поєднує математичні методи, алгоритми комп'ютерних наук та принципи фізики оптичних систем. В основі цієї дисципліни лежить розуміння цифрового зображення як дискретної функції інтенсивності, визначеної в просторовій області та представленої матрицею числових значень, кожне з яких відповідає окремому пікселю.

Концептуальна модель цифрового зображення базується на принципах дискретизації та квантування безперервного візуального сигналу. Процес дискретизації передбачає розбиття просторової області на регулярну сітку елементів, тоді як квантування забезпечує перетворення безперервних значень інтенсивності в дискретний набір рівнів. Ця двоетапна трансформація створює цифрове представлення, придатне для комп'ютерної обробки.

Теоретичні основи обробки зображень ґрунтуються на математичному апараті лінійної алгебри, теорії сигналів та математичного аналізу. Операції над зображеннями можуть бути класифіковані за різними критеріями: за типом перетворення (точкові, локальні, глобальні), за характером обробки (лінійні та нелінійні), за призначенням (покращення, відновлення, сегментація, аналіз). Кожна категорія операцій має свої специфічні математичні формулювання та алгоритмічні реалізації.

Фундаментальне значення для розуміння обробки зображень має концепція частотної області, що базується на теоремі Фур'є. Перетворення зображення з просторової області в частотну дозволяє виявити періодичні структури, здійснити фільтрацію шуму та реалізувати ефективні алгоритми

згортки. Дискретне перетворення Фур'є та його швидка реалізація відіграють ключову роль у багатьох алгоритмах обробки зображень.

## 1.2 Методи класичної обробки зображень

Класичні методи обробки зображень сформувалися на основі математично обґрунтованих підходів до аналізу та трансформації візуальної інформації. Ці методи характеризуються детермінованістю алгоритмів, прозорістю математичних операцій та можливістю точного контролю параметрів обробки.

Просторова фільтрація становить основу багатьох класичних алгоритмів обробки зображень. Лінійна просторова фільтрація базується на операції згортки зображення з ядром фільтра, що дозволяє реалізувати різноманітні ефекти: згладжування, загострення, виділення контурів. Гауссівський фільтр забезпечує оптимальне згладжування з мінімальними спотвореннями частотних характеристик, тоді як фільтр Лапласа ефективно виділяє області різких змін інтенсивності.

Морфологічна обробка зображень представляє окремий клас методів, заснований на теорії множин та топологічних перетвореннях. Базові морфологічні операції ерозії та дилатації дозволяють змінювати форму об'єктів на зображенні, тоді як їх комбінації (відкриття, закриття) забезпечують складніші перетворення структури зображення. Морфологічні методи особливо ефективні для обробки бінарних зображень та аналізу форми об'єктів.

Сегментація зображень в рамках класичних підходів базується на аналізі статистичних властивостей пікселів та їх просторових відношень. Методи порогової сегментації використовують гістограму зображення для визначення оптимальних значень поділу на області. Алгоритм Отсу забезпечує автоматичний вибір порогового значення на основі максимізації міжкласової дисперсії. Методи росту областей та водорозділу дозволяють

здійснювати сегментацію складних зображень з урахуванням просторової зв'язності пікселів.

Виділення ознак в класичних методах обробки зображень спирається на аналіз локальних властивостей зображення. Детектори кутів Харріса та SIFT (Scale-Invariant Feature Transform) забезпечують виявлення характерних точок, інваріантних до масштабування та поворотів. Дескриптори локальних бінарних патернів (LBP) та гістограм орієнтованих градієнтів (HOG) дозволяють створювати компактні представлення текстурних та структурних властивостей зображень.

### 1.3 Поняття і принципи роботи машинного навчання

Машинне навчання [1-7] як парадигма створення інтелектуальних систем базується на принципі автоматичного виявлення закономірностей в даних без явного програмування алгоритмів розв'язання конкретних задач. Цей підхід кардинально відрізняється від традиційного програмування, де логіка обробки інформації жорстко кодується програмістом. Дуже часто використовуються алгоритми нейронних мереж (рисунок 1.1, 1.2).



Рисунок 1.1 – Класифікація нейронних мереж

Теоретичні основи машинного навчання спираються на статистичну теорію навчання та теорію ймовірностей. Центральною концепцією є поняття узагальнення - здатності навченої моделі ефективно працювати з новими, раніше не баченими даними. Цей принцип формалізується через мінімізацію емпіричного ризику з регуляризаційними обмеженнями, що запобігають перенавчанню моделі.

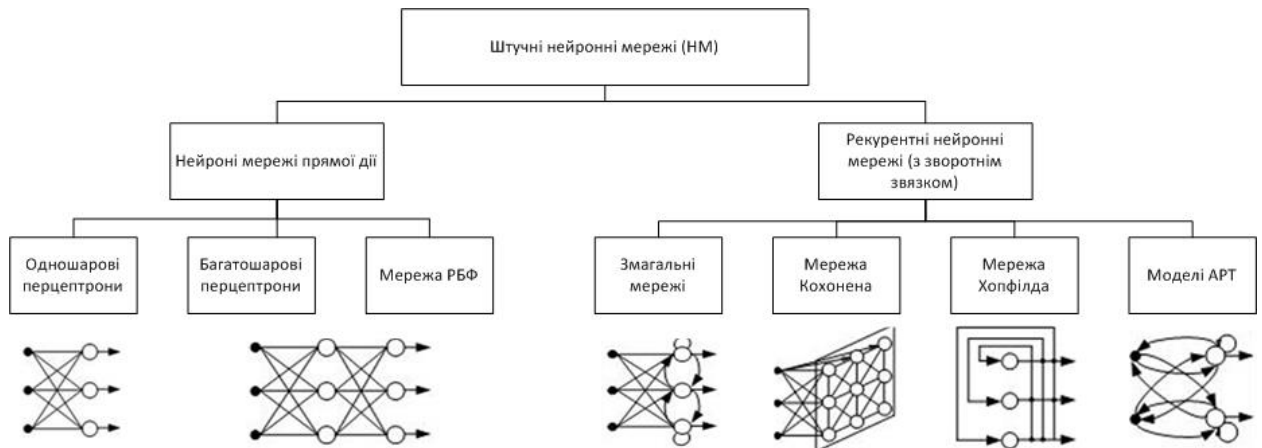


Рисунок 1.2 – Класифікація ШНМ

Процес машинного навчання може бути представлений як оптимізаційна задача, де метою є знаходження параметрів моделі, що мінімізують функцію втрат на навчальних даних. Вибір функції втрат залежить від типу задачі: для регресії зазвичай використовується середньоквадратична помилка, для класифікації - крос-ентропія або hinge loss. Процедура оптимізації може здійснюватися різними методами, від класичного градієнтного спуску до сучасних адаптивних алгоритмів.

Фундаментальним аспектом машинного навчання є проблема балансу між здатністю моделі до навчання (bias) та її чутливістю до випадкових флуктуацій в даних (variance). Цей компроміс визначає загальну ефективність моделі та вимагає ретельного налаштування складності моделі відносно обсягу та якості навчальних даних.

Валідація моделей в машинному навчанні передбачає використання статистично обґрунтованих методів оцінки якості. Крос-валідація дозволяє

отримати надійні оцінки ефективності моделі при обмежених обсягах даних. Метрики якості мають відповідати специфіці задачі: точність, повнота та F-міра для класифікації, середньоабсолютна та середньоквадратична помилки для регресії.

#### 1.4 Класифікація алгоритмів машинного навчання

Систематизація алгоритмів машинного навчання базується на декількох фундаментальних критеріях, що відображають різні аспекти процесу навчання та характер використовуваної інформації. Найбільш загальним є поділ за типом навчання: навчання з учителем, навчання без учителя та навчання з підкріпленням (рисунок 1.3).

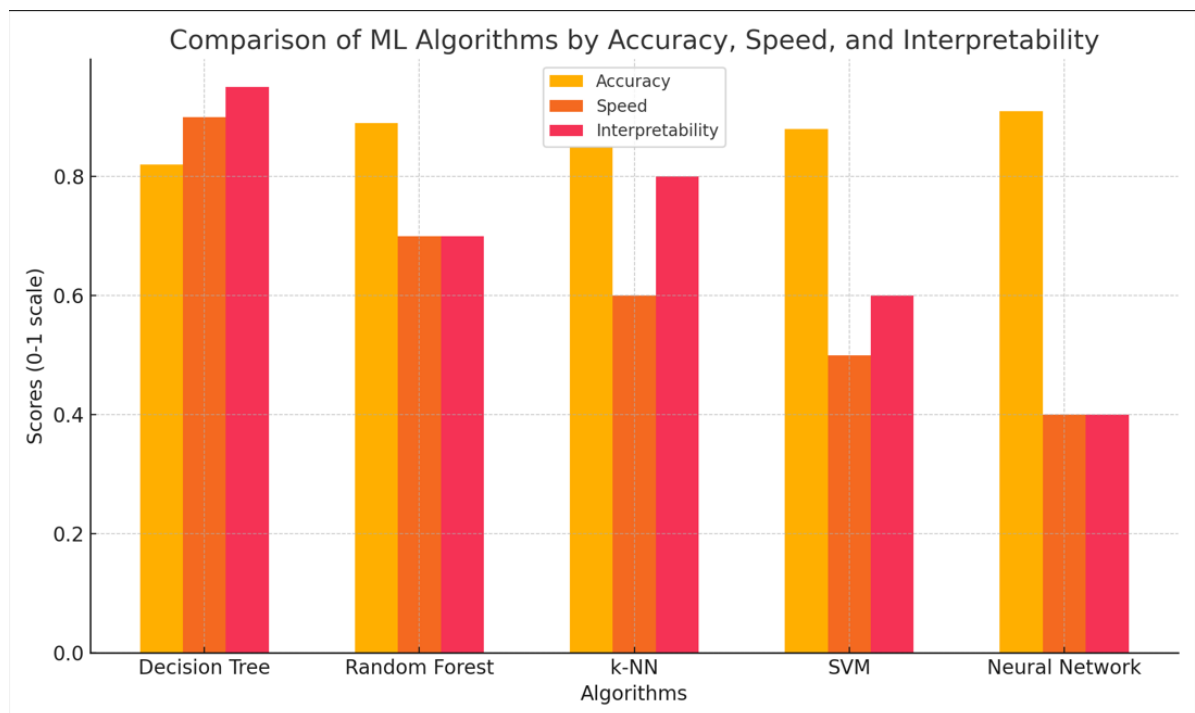


Рисунок 1.3 – Порівняльний аналіз методів машинного навчання

Навчання з учителем характеризується наявністю пар "вхід-вихід" в навчальних даних, що дозволяє моделі навчатися встановлювати відповідність між вхідними характеристиками та цільовими значеннями. В рамках цієї парадигми виділяються задачі класифікації, де мета полягає в

присвоєнні об'єктам дискретних міток класів, та регресії, спрямованої на передбачення безперервних числових значень. Класичними представниками цього підходу є лінійна та логістична регресія, методи опорних векторів, дерева рішень та ансамблеві методи.

Навчання без учителя працює з даними, що не містять цільових міток, та спрямоване на виявлення прихованих структур та закономірностей. Кластеризація дозволяє групувати схожі об'єкти без апріорної інформації про кількість та природу груп. Алгоритми зменшення розмірності, такі як метод головних компонент та t-SNE, забезпечують компактне представлення високовимірних даних зі збереженням найважливіших характеристик. Генеративні моделі дозволяють моделювати розподіл даних та генерувати нові зразки.

Навчання з підкріпленням базується на взаємодії агента з середовищем через систему винагород та покарань. Цей підхід особливо ефективний для задач послідовного прийняття рішень в умовах невизначеності. Q-навчання та policy gradient методи представляють основні напрямки розвитку цієї парадигми.

За характером обробки даних алгоритми можуть бути класифіковані як параметричні та непараметричні. Параметричні моделі мають фіксовану кількість параметрів незалежно від розміру навчальної вибірки, тоді як непараметричні моделі адаптують свою складність до обсягу даних. За способом навчання виділяються пакетні (batch) та онлайн алгоритми, що відрізняються стратегією обробки навчальних даних.

## 1.5 Використання нейронних мереж для обробки зображень

Нейронні мережі в контексті обробки зображень представляють собою біологічно-інспіровані обчислювальні моделі, що здатні автоматично вивчати ієрархічні представлення візуальної інформації. Фундаментальною перевагою нейромережевого підходу є можливість адаптивного формування

ознак безпосередньо з вихідних даних, що усуває необхідність ручного проектування дескрипторів.

Згорткові нейронні мережі (CNN) становлять основу сучасних методів обробки зображень завдяки своїй здатності ефективно моделювати просторові залежності та інваріантність до трансляцій. Архітектура CNN базується на чергуванні згорткових шарів, що здійснюють локальну обробку зображення, та шарів об'єднання (pooling), що забезпечують зменшення просторової розмірності та підвищення стійкості до малих деформацій.

Операція згортки в контексті нейронних мереж узагальнює класичну згортку додаванням можливості навчання параметрів фільтрів. Кожен згортковий шар містить множину фільтрів, що навчаються виявляти різні візуальні патерни: від простих контурів та текстур на нижніх рівнях до складних семантичних структур на вищих рівнях. Розмір рецептивного поля визначає область зображення, що впливає на активацію конкретного нейрона, та зростає з глибиною мережі.

Ієрархічна природа згорткових мереж забезпечує поступове підвищення рівня абстракції представлень. Початкові шари виявляють низькорівневі ознаки, такі як краї, кути та прості текстури. Проміжні шари комбінують ці елементарні ознаки для формування більш складних паттернів, що відповідають частинам об'єктів. Глибокі шари створюють високорівневі семантичні представлення цілих об'єктів та сцен.

Механізми регуляризації в нейронних мережах для обробки зображень включають dropout, batch normalization та data augmentation. Dropout запобігає перенавчанню шляхом випадкового вимкнення частини нейронів під час навчання. Batch normalization нормалізує входи кожного шару, що стабілізує процес навчання та дозволяє використовувати вищі швидкості навчання. Data augmentation збільшує різноманітність навчальних даних через застосування різноманітних трансформацій зображень.

## 2 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ОБРОБКИ ЗОБРАЖЕНЬ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

### 2.1 Огляд популярних нейронних архітектур

Еволюція архітектур (рисунок 2.1) згорткових нейронних мереж відображає прогрес у розумінні принципів ефективної обробки візуальної інформації. Кожна епохальна архітектура вносила концептуальні інновації, що стали основою для подальшого розвитку галузі.

AlexNet, представлена в 2012 році, стала першою глибокою згортковою мережею, що продемонструвала можливості GPU-обчислень для навчання великих нейронних мереж. Архітектура складалася з восьми шарів: п'яти згорткових та трьох повнозв'язних, і вперше використала ReLU активацію та dropout регуляризацію. Ключовим досягненням стало зменшення помилки класифікації на ImageNet більш ніж на 10% порівняно з попередніми методами.

VGG мережі запровадили принцип систематичного збільшення глибини мережі при використанні малих фільтрів розміром  $3 \times 3$ . Ця стратегія дозволила створити більш глибокі мережі (до 19 шарів) при збереженні обчислювальної ефективності. Архітектура VGG демонструє, що множина малих фільтрів може ефективно замінити великі фільтри, забезпечуючи більшу виразність при меншій кількості параметрів.

GoogleNet (Inception) революціонізувала проектування архітектур введенням inception блоків, що здійснюють паралельну обробку різними фільтрами в межах одного шару. Цей підхід дозволяє мережі автоматично вибирати оптимальний масштаб аналізу для різних частин зображення. Додатково, використання global average pooling замість повнозв'язних шарів суттєво зменшило кількість параметрів.

ResNet архітектура вирішила проблему деградації градієнтів в дуже

глибоких мережах введенням залишкових з'єднань (skip connections). Ці з'єднання дозволяють градієнтам проходити безпосередньо до нижніх шарів, забезпечуючи стабільне навчання мереж з сотнями шарів. Концепція залишкового навчання змінила розуміння оптимізації глибоких мереж та стала стандартом для багатьох сучасних архітектур.

Архітектура	Призначення	Ключові особливості
CNN	Обробка зображень, відео, просторових даних	Згорткові фільтри, локальність, ієрархічне вилучення ознак
RNN	Обробка послідовностей, часових рядів, мови	Зворотні зв'язки, пам'ять про попередні стани
GAN	Генерація нових даних, зокрема зображень	Архітектура з двома мережами: генератор і дискриміратор
Архітектура	Переваги	Недоліки
CNN	Висока точність при класифікації зображень	Не пристосована до обробки послідовностей
RNN	Добре працює з часовими залежностями	Складність тренування, проблема з градієнтами
GAN	Здатність створювати реалістичні нові дані	Важке налаштування, нестабільність під час навчання

Рисунок 2.1 – Огляд архітектур

DenseNet розвинула ідею skip connections, створюючи щільні з'єднання між всіма шарами в межах блоку. Кожен шар отримує входи від всіх попередніх шарів, що забезпечує максимальне повторне використання ознак та зменшує проблему зникаючих градієнтів. Ця архітектура досягає високої ефективності при меншій кількості параметрів.

Transformer архітектури, спочатку розроблені для обробки природної мови, знаходять застосування в комп'ютерному зорі через Vision Transformer (ViT). Механізм self-attention дозволяє моделювати довгі залежності в зображеннях без обмежень локальності, характерних для згорткових операцій. Цей підхід демонструє перспективи універсальних архітектур для різних типів даних.

## 2.2 Методи підготовки та попередньої обробки даних

Якість та представлення навчальних даних критично впливають на ефективність алгоритмів машинного навчання. Процес підготовки даних для обробки зображень включає множину етапів, кожен з яких вимагає ретельного аналізу та оптимізації відповідно до специфіки конкретної задачі.

Нормалізація зображень становить фундаментальний етап підготовки даних, що забезпечує стабільність та ефективність процесу навчання. Стандартизація значень пікселів до діапазону  $[0,1]$  або  $[-1,1]$  усуває проблеми, пов'язані з різними масштабами вхідних значень. Більш складні методи нормалізації включають Z-score стандартизацію з використанням статистик набору даних та histogram equalization для покращення контрасту.

Аугментація даних представляє потужний інструмент збільшення різноманітності навчальної вибірки без збору додаткових зображень. Геометричні трансформації включають повороти, відображення, масштабування та зсуви, що підвищують інваріантність моделі до просторових деформацій. Фотометричні перетворення змінюють яскравість, контраст, насиченість та відтінок, імітуючи варіації умов освітлення та характеристик камери.

Сучасні методи аугментації включають більш складні стратегії. CutOut та CutMix методи випадково маскують або змішують частини зображень, змушуючи модель фокусуватися на різноманітних ознаках. MixUp здійснює лінійну інтерполяцію між парами зображень та їх мітками, створюючи віртуальні навчальні приклади. AutoAugment використовує автоматичний пошук оптимальних стратегій аугментації для конкретних наборів даних.

Сегментація та вирізання об'єктів інтересу дозволяють сфокусувати увагу моделі на релевантних частинах зображення. Методи automatic cropping базуються на детекції об'єктів або аналізі уваги для визначення найбільш інформативних регіонів. Smart cropping техніки враховують композиційні правила та семантичний вміст для створення естетично привабливих та інформативно багатих фрагментів.

Балансування класів в наборах даних критично важливе для навчання неупереджених моделей. Методи oversampling збільшують представлення рідкісних класів через дублювання або синтетичну генерацію зразків. Undersampling зменшує кількість зразків домінуючих класів. Weighted loss functions присвоюють різні ваги класам під час навчання, компенсуючи

дисбаланс без зміни структури даних.

Попередня обробка для конкретних задач може включати спеціалізовані техніки. Для медичних зображень застосовуються методи підвищення контрасту та шумоподавлення. Для супутникових знімків використовуються атмосферні корекції та спектральні перетворення. Для зображень низької якості застосовуються методи *super-resolution* та *denoising*.

### 2.3 Вибір та оптимізація гіперпараметрів моделей

Гіперпараметри нейронних мереж визначають архітектурні характеристики та параметри процесу навчання, що не оновлюються автоматично під час тренування. Оптимальний вибір цих параметрів критично впливає на ефективність моделі та вимагає систематичного підходу до пошуку та валідації.

Архітектурні гіперпараметри включають кількість шарів, розміри фільтрів, кількість каналів та типи активаційних функцій. Глибина мережі визначає її виразну здатність, але надмірне збільшення може призвести до проблем оптимізації та перенавчання. Ширина шарів (кількість каналів) впливає на здатність мережі моделювати складні залежності, але також збільшує обчислювальні витрати.

Гіперпараметри навчання контролюють процес оптимізації параметрів моделі. Швидкість навчання (*learning rate*) є найкритичнішим параметром, що визначає розмір кроків при оновленні ваг. Занадто висока швидкість може призвести до нестабільності навчання, тоді як занадто низька сповільнює конвергенцію. Стратегії адаптивної зміни швидкості навчання включають *step decay*, *exponential decay* та *cosine annealing*.

Розмір пакету (*batch size*) впливає на стабільність градієнтів та швидкість навчання. Великі пакети забезпечують стабільніші оцінки градієнтів, але вимагають більше пам'яті та можуть призвести до гіршої генералізації. Малі пакети вносять шум в процес оптимізації, що може

допомогти уникнути локальних мінімумів, але сповільнює навчання.

Регуляризаційні гіперпараметри контролюють складність моделі та запобігають перенавчанню. Параметр L2 регуляризації (weight decay) штрафує великі значення ваг, сприяючи більш простим моделям. Ймовірність dropout визначає частку нейронів, що випадково відключаються під час навчання. Параметри batch normalization впливають на стабілізацію розподілів активацій.

Методи автоматичного пошуку гіперпараметрів дозволяють систематично досліджувати простір можливих конфігурацій. Grid search здійснює вичерпний пошук по заданій сітці значень, але стає неефективним при великій кількості параметрів. Random search показує кращу ефективність для високорозмірних просторів параметрів завдяки кращому покриттю важливих напрямків.

Bayesian optimization використовує ймовірнісні моделі для побудови surrogate функцій якості та інтелектуального вибору наступних точок для оцінки. Цей підхід особливо ефективний для дорогих обчислень, характерних для навчання глибоких мереж. Genetic algorithms та particle swarm optimization представляють альтернативні метаевристичні підходи до оптимізації гіперпараметрів.

Population-based training дозволяє одночасно навчати множину моделей з різними гіперпараметрами та адаптивно змінювати конфігурації на основі проміжних результатів. Цей підхід поєднує переваги паралельного пошуку з можливістю динамічної адаптації параметрів під час навчання.

## 2.4 Програмні засоби і бібліотеки для роботи з зображеннями

Сучасна екосистема програмних засобів для машинного навчання та обробки зображень характеризується різноманітністю спеціалізованих інструментів, кожен з яких оптимізований для конкретних аспектів розробки та дослідження. Вибір відповідного програмного стеку критично впливає на

продуктивність розробки та ефективність рішень (рисунок 2.2).

TensorFlow представляє комплексну платформу для машинного навчання, що забезпечує інструменти від дослідницького прототипування до промислового розгортання. Keras API надає високорівневий інтерфейс для швидкого проектування та експериментування з нейронними мережами. TensorFlow Extended (TFX) забезпечує повний pipeline для промислових ML систем, включаючи валідацію даних, тренування моделей та моніторинг в продакшені.

Фреймворк	Основне призначення	Сумісність з Google Colab
TensorFlow	Глибоке навчання, розробка нейронних мереж	Повна інтеграція, попередньо встановлений
PyTorch	Глибоке навчання, наукові дослідження та візуалізація	Повна інтеграція, попередньо встановлений
OpenCV	Обробка зображень, комп'ютерний зір	Повна інтеграція, попередньо встановлений
Фреймворк	Переваги	Недоліки
TensorFlow	Підтримка мобільних та веб-платформ, TPU	Складніша відлагодження, громіздкий синтаксис
PyTorch	Гнучкість, зручність для дослідників, динамічне обчислення	Менш зручний у виробничих середовищах
OpenCV	Швидкість, простота базових операцій зображень	Не призначений для глибокого навчання

Рисунок 2.2 – Огляд середовищ розробки

PyTorch відзначається гнучкістю та інтуїтивністю, особливо в дослідницьких завданнях. Dynamic computation graphs дозволяють змінювати архітектуру мережі під час виконання, що спрощує реалізацію складних моделей. TorchVision надає предтренувані моделі та стандартні датасети для комп'ютерного зору. PyTorch Lightning абстрагує boilerplate код, дозволяючи зосередитися на логіці моделі.

OpenCV становить фундаментальну бібліотеку для класичної обробки зображень та комп'ютерного зору. Багатий набір функцій включає фільтрацію, морфологічні операції, детекцію особливостей та калібрування камер. Оптимізовані реалізації алгоритмів забезпечують високу продуктивність для real-time додатків. Підтримка різних мов програмування робить OpenCV універсальним інструментом.

Scikit-image надає Python-інтерфейс для наукової обробки зображень з акцентом на простоту використання та якість документації. Бібліотека включає алгоритми сегментації, аналізу форми, вимірювання властивостей

об'єктів та морфологічних операцій. Інтеграція з SciPy ecosystem забезпечує seamless робочий процес для наукових досліджень.

Albumentations спеціалізується на ефективній аугментації зображень для завдань машинного навчання. Оптимізовані реалізації трансформацій забезпечують високу швидкість обробки великих датасетів. Підтримка різних типів анотацій (bounding boxes, masks, keypoints) з автоматичною трансформацією спрощує роботу з задачами детекції та сегментації.

PIL/Pillow забезпечує базову функціональність для роботи з зображеннями в Python, включаючи завантаження, збереження та прості трансформації. Matplotlib надає потужні інструменти візуалізації результатів та аналітики. NumPy служить фундаментом для числових обчислень та представлення зображень як багатовимірних масивів.

Спеціалізовані інструменти включають NVIDIA DALI для високопродуктивного завантаження та попередньої обробки даних, Detectron2 для задач детекції об'єктів, MMDetection для комплексного toolkit детекції та сегментації. Cloud-based рішення такі як Google Cloud Vision API та Amazon Rekognition надають готові сервіси для типових задач комп'ютерного зору.

### 3 РЕАЛІЗАЦІЯ СИСТЕМИ ОБРОБКИ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

У контексті розробки комп'ютерної системи обробки зображень із використанням технологій глибокого навчання важливим етапом є чітке формулювання задачі, що вирішується, а також обґрунтування вибору відповідного підходу для її реалізації.

У даній роботі розглядається задача автоматичної класифікації зображень, яка передбачає присвоєння кожному вхідному зображенню одного з фіксованого набору класів. Така задача є базовою, але водночас репрезентативною для широкого спектра прикладних застосувань: від медичної діагностики та агроскаутигу до систем відеоспостереження та автоматичного маркування вмісту.

Аналіз існуючих методів показав, що традиційні алгоритми машинного навчання (наприклад, метод опорних векторів, дерева рішень, наївний байєсівський класифікатор) вимагають складного етапу ручного проектування ознак, який є вкрай чутливим до якості вхідних даних і не гарантує високої узагальнювальної здатності моделі. Натомість глибокі нейронні мережі, зокрема згорткові нейронні мережі, демонструють здатність до автоматичного виділення релевантних ознак без втручання людини, що робить їх ефективним інструментом для обробки візуальної інформації.

Постановка задачі передбачає побудову моделі глибокого навчання, здатної з високою точністю класифікувати зображення за наперед визначеними категоріями. Для реалізації буде використано публічно доступний датасет CIFAR-10, що містить 60 000 кольорових зображень розміром  $32 \times 32$  пікселі, розподілених по 10 класах. Такий вибір обумовлений балансом між складністю задачі та її придатністю для навчання в хмарному середовищі Google Colab із обмеженими обчислювальними ресурсами.

### 3.1 Процес розробки КС

Процес розробки комп'ютерної системи обробки зображень із використанням глибинного навчання включає проектування архітектури моделі, підготовку вхідних даних, реалізацію алгоритмів навчання та валідації, а також організацію інтерактивного середовища для експериментів і тестування.

У якості інструментального середовища обрано платформу Google Colab, яка надає доступ до обчислювальних ресурсів GPU та TPU, а також забезпечує інтеграцію з фреймворками TensorFlow і Keras, що істотно спрощує реалізацію глибоких нейронних мереж.

Першим етапом розробки є підготовка вхідних даних. Датасет CIFAR-10 завантажується з вбудованої бібліотеки TensorFlow, після чого проходить стандартну обробку: масштабування значень пікселів до діапазону  $[0, 1]$  та перетворення міток у відповідний числовий формат. Задля підвищення узагальнювальної здатності моделі застосовується аугментація даних, зокрема випадкові горизонтальні віддзеркалення та зсуви зображень.

Архітектура моделі базується на згортковій нейронній мережі з кількома шарами. До її складу входять згорткові шари для виділення локальних ознак, шари підвибірки (max-pooling) для зменшення розмірності, а також повнозв'язні шари для остаточної класифікації. У якості функції активації використано ReLU (Rectified Linear Unit), яка забезпечує ефективну нелінійність, а фінальний шар має функцію softmax для багатокласової класифікації.

Навчання моделі здійснюється з використанням оптимізатора Adam, що поєднує переваги адаптивної швидкості навчання та моментуму. Функцією втрат обрано категорійну крос-ентропію, яка є стандартною для задач багатокласової класифікації. В процесі тренування фіксуються метрики точності як на тренувальній, так і на валідаційній вибірках, що дозволяє контролювати якість навчання та уникати перенавчання.

Після завершення навчання модель зберігається у форматі .h5 для подальшого використання або імпорту. На завершальному етапі проводиться візуалізація результатів класифікації, побудова графіків точності та втрат, а також аналіз неправильно класифікованих зображень для оцінки обмежень моделі.

Результатом є прототип функціональної системи, який демонструє здатність до автоматизованого розпізнавання візуальних образів на основі глибокого навчання, з можливістю масштабування під інші набори даних та адаптації до складніших сценаріїв обробки зображень.

### 3.2 Архітектура системи

Архітектура комп'ютерної системи обробки зображень базується на модульному принципі, що забезпечує гнучкість у налаштуванні, модифікації та масштабуванні її компонентів. Основними складовими є: модуль підготовки даних, модуль побудови й навчання моделі, модуль оцінки якості, а також інтерфейс користувача у вигляді інтерактивного середовища Google Colab.

На першому рівні архітектури розташовується модуль введення та попередньої обробки даних, який відповідає за завантаження зображень, їх масштабування, нормалізацію та аугментацію. Цей модуль гарантує, що дані, які надходять до нейронної мережі, мають уніфікований формат і не містять спотворень, здатних негативно вплинути на навчання.

Другим ключовим елементом є модуль побудови моделі, який реалізує згорткову нейронну мережу (CNN) з чітко визначеною структурою: послідовність згорткових і pooling-шарів для автоматичного вилучення ознак, завершена повнозв'язними шарами для класифікації. Така архітектура дозволяє моделі самостійно навчитися розпізнавати характерні патерни в зображеннях, зокрема краї, текстури, форми тощо.

Наступним етапом є модуль навчання, у якому відбувається

оптимізація ваг моделі за допомогою алгоритму градієнтного спуску з адаптивним кроком (Adam). У цьому ж модулі реалізовано контроль за якістю навчання через валідаційну вибірку, що дозволяє виявити перенавчання або недостатнє узагальнення.

Четвертий компонент архітектури – модуль оцінки продуктивності, який виконує обчислення ключових метрик (точність, precision, recall, F1-score), побудову кривих навчання, а також візуалізацію результатів класифікації. Цей модуль забезпечує користувача зворотним зв'язком щодо ефективності моделі.

Нарешті, інтерфейс користувача, реалізований у середовищі Google Colab, поєднує всі компоненти в єдину систему. Його основною перевагою є доступність, можливість інтерактивної взаємодії та наочна демонстрація результатів роботи моделі.

Таким чином, архітектура системи орієнтована на простоту використання, модульність і високу ефективність обробки зображень за рахунок інтеграції сучасних інструментів глибокого навчання.

### 3.3 Вибір моделі

Одним із найважливіших етапів побудови системи обробки зображень є вибір архітектури моделі глибокого навчання, що найбільш повно відповідає специфіці поставленої задачі. У межах цієї роботи основною задачею є класифікація зображень, що обумовлює доцільність використання згорткових нейронних мереж (CNN – Convolutional Neural Networks), які спеціалізуються на роботі з просторовими даними та демонструють найкращі результати у завданнях комп'ютерного зору.

CNN є природним вибором для аналізу зображень, оскільки її архітектура враховує локальні просторові залежності пікселів. Завдяки згортковим шарам мережа здатна автоматично виявляти ознаки, такі як краї, кути, текстури та форми, поступово формуючи високорівневі абстракції. Це

дозволяє моделі ефективно навчатись на великих наборах даних без потреби в ручному визначенні ознак.

У роботі було розглянуто кілька варіантів архітектур: базова CNN, ResNet (ResNet-18/34) та U-Net. Остаточний вибір зупинено на базовій CNN через її простоту, швидкість навчання та достатню ефективність на задачах класифікації невеликих зображень, як у датасеті CIFAR-10. Використання ResNet, яка реалізує залишкові зв'язки (residual connections) та дозволяє будувати глибші мережі без втрати якості, було визнано доцільним лише у випадках роботи з більшими або складнішими наборами даних. Що стосується архітектури U-Net, вона орієнтована переважно на задачі семантичної сегментації, а не класифікації, тому була виключена з переліку кандидатів.

Таким чином, вибір моделі продиктований співвідношенням між складністю задачі, обчислювальними можливостями середовища (Google Colab), а також потребою в інтерпретованості результатів. Базова згортова нейронна мережа забезпечує баланс між точністю, швидкістю та простотою реалізації, що робить її оптимальним рішенням для поставленої задачі.

### 3.4 Обробка та підготовка даних

Ефективність роботи моделей глибинного навчання у значній мірі залежить від якості та репрезентативності вхідних даних. Невідповідна підготовка зображень може призвести до поганого узагальнення, переобучення або значного зниження точності класифікації. Тому етап попередньої обробки та аугментації даних є невід'ємною частиною побудови будь-якої системи комп'ютерного зору.

Першим кроком є нормалізація зображень – процес приведення значень пікселів до уніфікованого діапазону. У цій роботі було використано масштабування значень з діапазону  $[0, 255]$  до  $[0, 1]$ , що дозволяє прискорити процес навчання та зробити його більш стабільним, уникнувши

перенасичення градієнтів у мережі.

Другим важливим кроком стала аугментація даних, тобто штучне збільшення обсягу навчальної вибірки шляхом створення варіацій існуючих зображень. Такий підхід дозволяє підвищити здатність моделі до узагальнення, імітуючи зміни, які можуть зустрічатися у реальних умовах. Зокрема, було застосовано випадкове горизонтальне віддзеркалення, зсув зображень у межах 10% від розміру кадру, а також легкі зміни яскравості. Ці трансформації реалізовано за допомогою інструментів бібліотеки Keras (ImageDataGenerator), що забезпечує інтеграцію з навчальним циклом у реальному часі.

Крім того, всі зображення були приведені до єдиного розміру  $32 \times 32$  пікселі, відповідно до формату датасету CIFAR-10. Це дозволяє уникнути проблем з форматом вхідних даних та забезпечує узгодженість під час навчання.

Таким чином, обробка та підготовка даних відіграють ключову роль у забезпеченні високої точності та надійності моделі, дозволяючи покращити її здатність до генералізації навіть за наявності обмеженої кількості навчальних прикладів.

### 3.5 Реалізація в Google Colab

Для реалізації комп'ютерної системи обробки зображень було обрано інтерактивне хмарне середовище Google Colab, що забезпечує доступ до апаратного прискорення (GPU), підтримує бібліотеки TensorFlow та Keras, а також дозволяє здійснювати повноцінний науково-дослідний цикл у межах одного ноутбука – від завантаження даних до візуалізації результатів.

У рамках проєкту використовувався публічний набір даних CIFAR-10, який містить 60 000 зображень розміром  $32 \times 32$  пікселі, рівномірно розподілених між 10 категоріями: літаки, автомобілі, птахи, коти, олені, собаки, жаби, коні, кораблі та вантажівки. Дані було завантажено з

вбудованого API TensorFlow (`tf.keras.datasets.cifar10.load_data()`), після чого проведено нормалізацію значень пікселів та кодування міток.

```

✓ 3s ▶ y_pred = model.predict(x_test)
      y_pred_classes = np.argmax(y_pred, axis=1)
      print(classification_report(y_test, y_pred_classes))
      print(confusion_matrix(y_test, y_pred_classes))

```

313/313 ————— 3s 10ms/step

	precision	recall	f1-score	support
0	0.75	0.73	0.74	1000
1	0.74	0.88	0.80	1000
2	0.63	0.58	0.60	1000
3	0.56	0.46	0.50	1000
4	0.69	0.59	0.64	1000
5	0.70	0.45	0.55	1000
6	0.65	0.86	0.74	1000
7	0.71	0.78	0.74	1000
8	0.76	0.82	0.79	1000
9	0.72	0.79	0.75	1000
accuracy				0.69 10000
macro avg				0.69 10000
weighted avg				0.69 10000

```

[[733  35  44  10  7  2  14  10  97  48]
 [ 11 882  1  2  1  2  11  2  19  69]
 [ 63  15 578  43  69  38  93  47  26  28]
 [ 23  29  81 461  57  90 121  62  32  44]
 [ 24  13  77  38 589  19 121  90  21  8]
 [ 23  19  67 199  55 449  64  85  13  26]
 [ 10  13  36  33  18  5 857  10  7  11]
 [ 17  11  25  25  49  32  16 779  11  35]
 [ 57  53  10  7  5  1  2  5 819  41]
 [ 19 122  4  8  3  5  15  4  29 791]]

```

Рисунок 3.1 – Результати оцінювання якості роботи

Рисунок 3.1 ілюструє результати оцінювання якості роботи згорткової нейронної мережі після завершення етапу навчання на тестовій вибірці. У верхній частині зображено класифікаційний звіт, який відображає ключові метрики продуктивності моделі для кожного з десяти класів: точність (*precision*), повноту (*recall*) та інтегральний показник F1. Всі значення розраховані на основі порівняння передбачених міток із фактичними, що дозволяє зробити висновки про узагальнювальну здатність моделі.

На рисунку 3.2 представлено графіки точності та втрат моделі в процесі навчання. Зліва видно стабільне зростання як тренувальної, так і валідаційної точності, що досягає близько 70%, що свідчить про поступове покращення якості класифікації. Справа зображено зниження функції втрат, причому валідаційні втрати зменшуються швидше та досягають нижчого рівня, ніж тренувальні. Така динаміка вказує на відсутність перенавчання та на те, що модель ефективно узагальнює інформацію з навчальних даних.

З аналізу звіту випливає, що модель демонструє достатньо стабільну роботу в межах кількох класів, зокрема тих, які мають яскраво виражені візуальні ознаки. Наприклад, для класів із вищими значеннями recall та F1-міри, спостерігається хороша здатність моделі правильно розпізнавати вхідні приклади. Водночас, наявність значного розриву між точністю та повнотою у ряді інших класів свідчить про нерівномірне навчання або зміщення моделі до певних категорій. Особливо це помітно для класів, де F1-міра не перевищує позначки 0.55, що вказує на труднощі в розпізнаванні слабо виражених ознак або візуально подібних зображень.

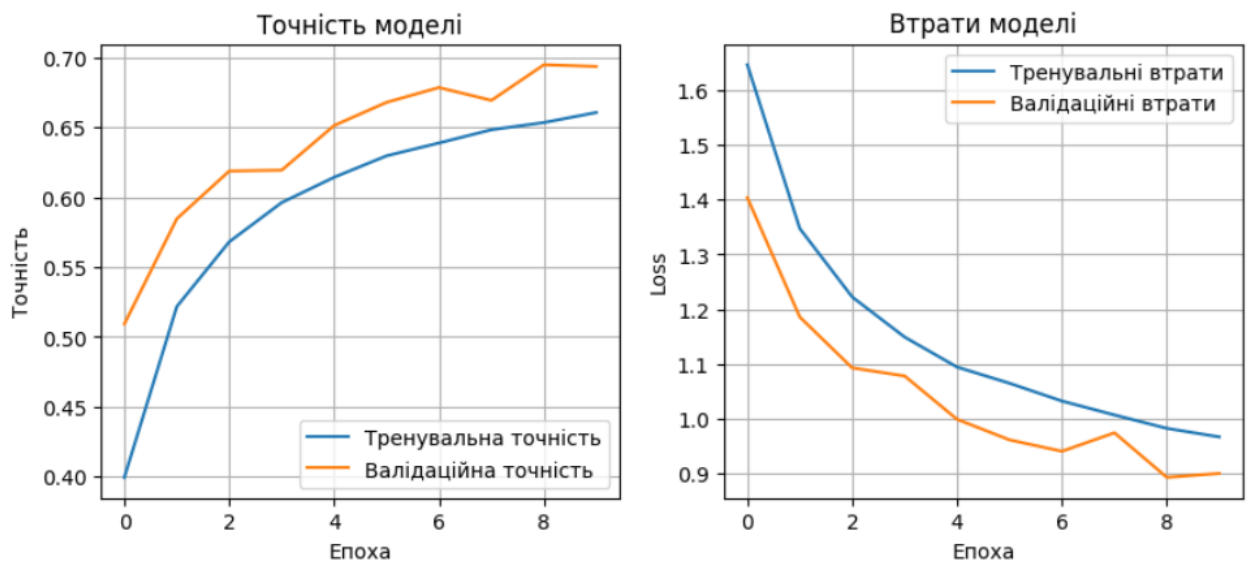


Рисунок 3.2 – Аналіз результатів

У нижній частині рисунка представлено матрицю плутанини, яка деталізує кількісний розподіл правильних та хибних передбачень моделі.

Діагональні значення відображають кількість правильно класифікованих прикладів для кожного класу, тоді як позадіагональні свідчать про помилкові співставлення. Наявність значного числа помилок у певних клітинках, особливо в контексті класів із низьким значенням recall, дозволяє визначити напрямки, у яких модель припускається найбільше хибних рішень. Це може бути пов'язано як із неоднозначністю вхідних даних, так і з недостатньою глибиною мережі для захоплення складних просторових закономірностей.

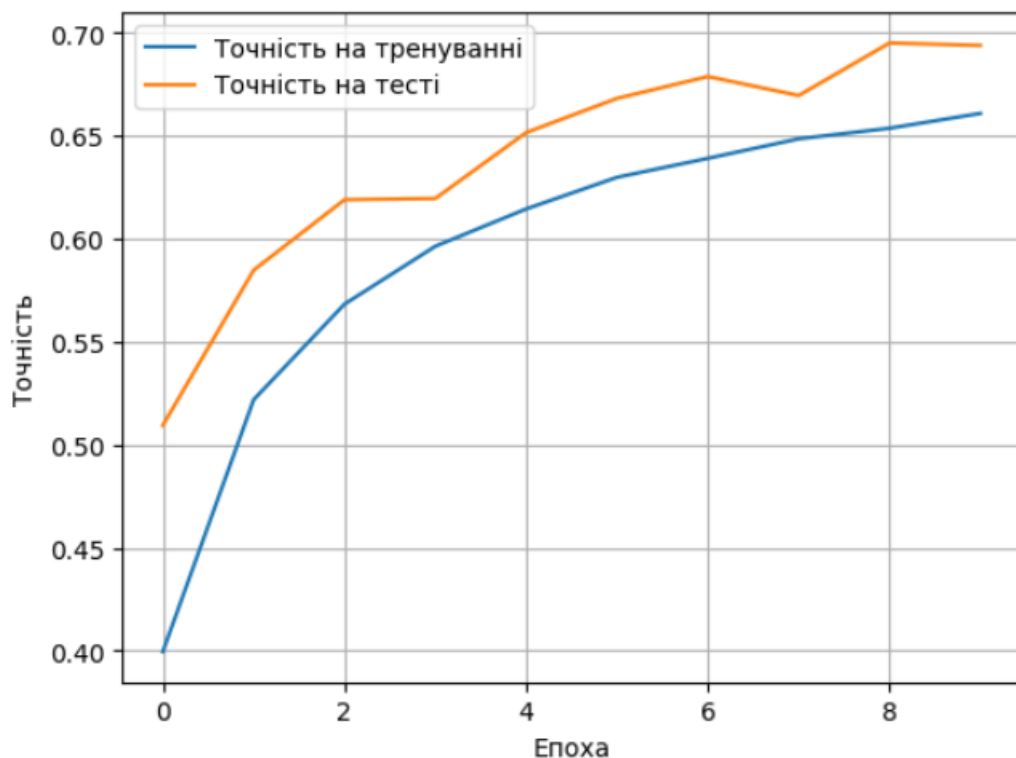


Рисунок 3.3 – Аналіз результатів

На графіку рисунка 3.3 зображено зміну точності моделі під час навчання та тестування протягом десяти епох. Валідаційна (тестова) точність зростає швидше, ніж тренувальна, та стабілізується на рівні близько 70%, що перевищує тренувальні показники. Така поведінка свідчить про добру здатність моделі до узагальнення та відсутність явного перенавчання на навчальних даних. Модель демонструє стійке поліпшення якості класифікації з кожною епохою, що підтверджує ефективність обраної архітектури та гіперпараметрів.

Загальна точність моделі на тестовій вибірці становить 69%, що є прийнятним результатом для базової архітектури, однак вказує на наявність потенціалу для її вдосконалення. Зокрема, поліпшення можна досягти шляхом поглиблення моделі, впровадженням залишкових зв'язків або використанням більш збалансованих аугментаційних стратегій під час підготовки даних.

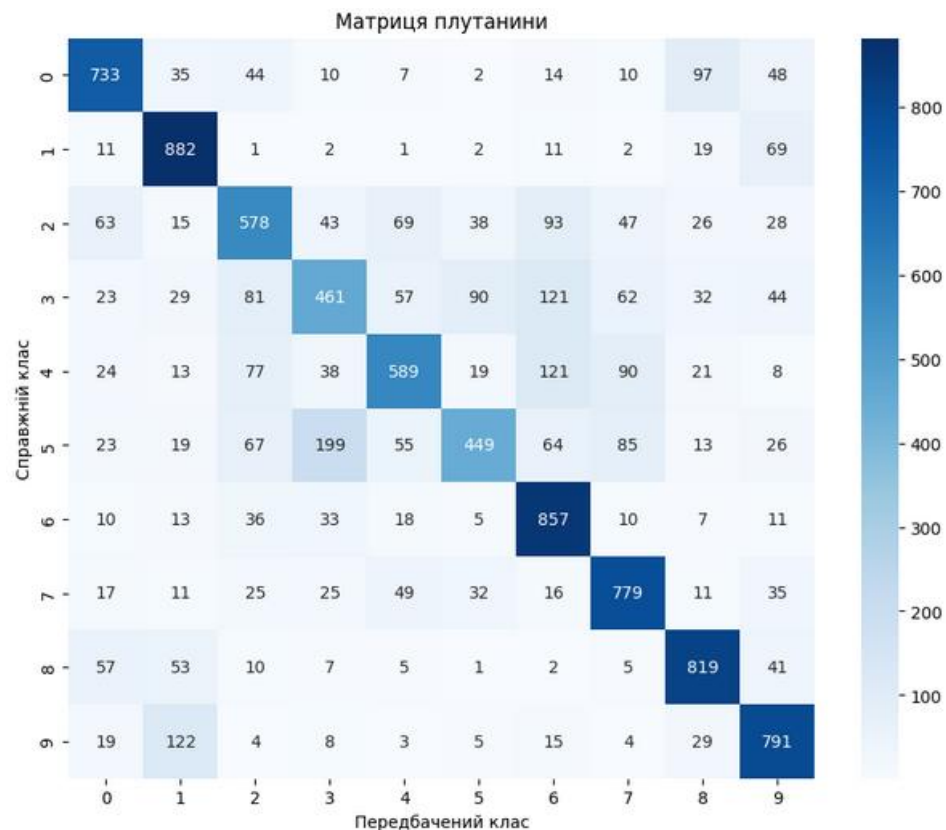


Рисунок 3.4 – Матриця плутанини

Рисунок 3.4 відображає матрицю плутанини для моделі класифікації, побудованої на основі згорткової нейронної мережі. По вертикалі вказані справжні класи тестових прикладів, тоді як по горизонталі – класи, які модель передбачила. Інтенсивність кольору в клітинках свідчить про кількість прикладів, що належать до відповідної комбінації справжнього та передбаченого класів.

Аналіз діагональних елементів дозволяє оцінити загальну якість класифікації: що більшими є значення вздовж діагоналі, то більше прикладів

було правильно класифіковано. Зокрема, моделі вдалося з високою точністю розпізнати приклади класів 1, 6, 8 та 9, що підтверджується високими значеннями на відповідних перетинах. Водночас помітною є наявність значної кількості хибних передбачень для класів 3 та 5, де спостерігається широка дисперсія значень за межами діагоналі. Це свідчить про труднощі моделі у відокремленні цих класів від інших, ймовірно через візуальну подібність або недостатню інформативність вхідних ознак.

Таким чином, представлена матриця плутанини дозволяє не лише кількісно оцінити точність класифікації, а й якісно виявити напрямки для подальшого вдосконалення моделі, зокрема через підсилення здатності розрізняти візуально схожі об'єкти.

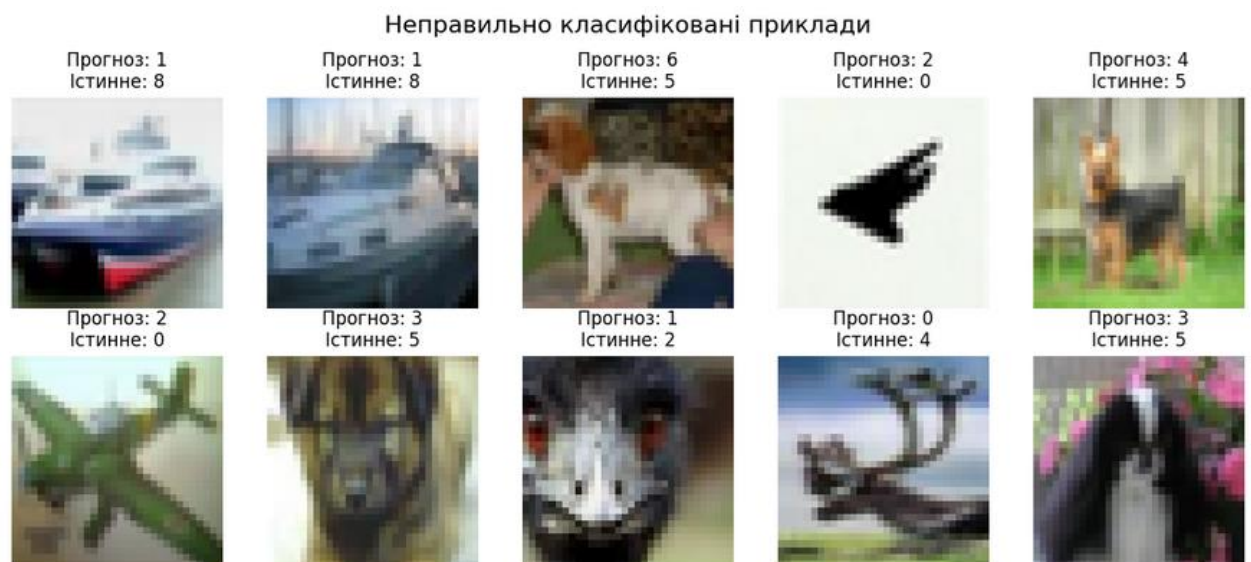


Рисунок 3.5 – Аналіз результатів

Архітектура моделі реалізована на основі згорткової нейронної мережі з наступними шарами: два згорткові шари з фільтрами 32 та 64, шари max pooling, flatten-операція та два повнозв'язні шари, з яких останній містить 10 нейронів з функцією активації softmax. Для оптимізації використано алгоритм Adam, функцією втрат – sparse\_categorical\_crossentropy.

Навчання проводилося протягом 10 епох з використанням валідаційної вибірки для контролю за якістю навчання. Аугментація даних здійснювалась

у режимі реального часу за допомогою генератора зображень Keras.

Оцінювання точності виконувалося як на основі стандартної метрики accuracy, так і з використанням розширених показників якості: precision, recall та F1-score. Для цього застосовано бібліотеку sklearn.metrics, яка дозволяє побудувати звіт класифікації по кожному класу та створити матрицю плутанини (confusion matrix), що візуалізує розподіл правильно та неправильно класифікованих зображень.

На рисунку 3.5 представлено приклади неправильних класифікацій, здійснених моделлю на тестовій вибірці. Над кожним зображенням зазначено передбачений клас («Прогноз») та його істинне значення («Істинне»), що дозволяє наочно оцінити помилки моделі.

Виявлені приклади демонструють характерні ситуації, у яких нейронна мережа зазнає труднощів з розпізнаванням. У деяких випадках це пов'язано з низькою якістю або розмитістю зображень, що ускладнює виділення ключових ознак. В інших ситуаціях причиною помилок є високий рівень візуальної подібності між об'єктами різних класів. Наприклад, кораблі були помилково віднесені до автомобілів, а собаки – до кішок або оленів, що свідчить про часткове перекриття ознак між цими категоріями.

Також помітно, що помилки моделі не є випадковими: вони зосереджені навколо певних класів, які мають більшу ймовірність плутанини. Це підтверджує попередні спостереження з аналізу матриці плутанини й вказує на системну природу слабких місць моделі. Такі візуалізації є важливими для діагностики якості навчання, оскільки дозволяють не лише кількісно оцінити похибки, а й інтерпретувати їх візуально.

В цілому, подібний аналіз є корисним інструментом для подальшого вдосконалення архітектури моделі або стратегії підготовки даних, зокрема шляхом включення додаткових ознак, посилення аугментації або балансування складних для класифікації класів.

Для наочного представлення результатів були побудовані графіки залежності точності та втрат від кількості епох, візуалізовано приклади

зображень з передбаченими класами, а також ідентифіковано помилки класифікації. Це дозволило зробити висновки про сильні та слабкі сторони моделі, а також оцінити її потенціал до узагальнення.

Таким чином, реалізація системи в Google Colab продемонструвала ефективність глибокого навчання у задачах класифікації зображень та підтвердила доцільність використання CNN для обробки візуальних даних у хмарному середовищі.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розроблено та реалізовано комп'ютерну систему обробки зображень із використанням технологій глибинного навчання, зокрема згорткових нейронних мереж. Робота охоплює повний цикл побудови моделі – від аналізу предметної області й постановки задачі до реалізації, навчання, тестування та візуалізації результатів у середовищі Google Colab.

Проведене дослідження підтвердило доцільність використання CNN у задачах класифікації зображень. Реалізована модель досягла прийняттого рівня точності на тестовій вибірці та продемонструвала здатність до узагальнення при обмеженій кількості навчальних прикладів. Було встановлено, що правильна підготовка даних – нормалізація, аугментація – істотно впливає на якість результатів. Візуалізація роботи моделі дала змогу виявити помилки класифікації, які, у свою чергу, дозволили зробити обґрунтовані висновки про можливі напрями подальшого вдосконалення.

Практична цінність розробленої системи полягає в її модульності, відкритості коду, простоті інтеграції в інші проєкти та адаптивності до нових наборів зображень. Окрім того, реалізація в середовищі Google Colab робить її доступною для широкого кола користувачів без потреби у спеціалізованому обладнанні.

У подальшому перспективним є розширення функціональності системи за рахунок використання глибших або спеціалізованих архітектур (ResNet, EfficientNet), впровадження трансформерів у комп'ютерний зір (Vision Transformer) або адаптація до задач сегментації та детекції об'єктів, що дозволить розширити сферу практичного застосування моделі.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Flach P. A. Machine Learning: The Art and Science of Algorithms that Makes Sense of Data. Cambridge: Cambridge University Press, 2012. 291 p. <https://doi.org/10.1017/CBO9780511973000> .
2. Phua, C. et al. A Comprehensive Survey of Data Mining-Based Fraud Detection Research. Artificial Intelligence Review, 55, 2021. P. 1985-2033.
3. C. Romero, S. Ventura. Educational data mining and learning analytics: An updated survey. Cornell University. Computer Science, 2024. 30 p. <https://doi.org/10.48550/arXiv.2402.07956> .
4. Q. Wang. Research on the Application of Machine Learning in Financial Anomaly Detection. iBusiness, 16, 2024. 173-183. <https://doi.org/10.4236/ib.2024.164012> .
5. О.Д. Земляний, О.Г. Байбуз. Огляд методів аналізу даних та методів машинного навчання при прогнозуванні ішемічної хвороби серця. Актуальні проблеми автоматизації та інформаційних технологій, т.27, 2023. С. 109-129.
6. E. Abdallah, Esraa A. Elsoud, A. Abdallah. A Survey of Data Mining Techniques for Digital Forensic Analysis. Procedia Computer Science, Volume 257, 2025. P. 731-736. <https://doi.org/10.1016/j.procs.2025.03.094>
7. A.Rotty, T.Dewayana, A.Habyba. Cross-Industry Standard Process for Data Mining (CRISP-DM) Approach in Determining the Most Significant Employee Engagement Drivers to Sales at X Car Dealership. IEOM Society International, 2023. P. 3368-2279. <https://doi.org/10.46254/AP03.20220552> .