

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра Інформатики

**ЗВІТ**  
**ПРО ПРАКТИКУ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ**  
в період з 30 вересня по 16 листопада 2024 року

Тема індивідуального завдання:

**Дослідження використання нейромереж для класифікації зображень**

Звіт розробив  
ст. гр. ІНФМ-23-2  
Яковенко В. В.  
Залікова книжка  
№ 23.122ІНФМ.0222

Керівник практики  
доц. Творошенко І. С.  
Робота захищена із оцінкою  
\_\_\_\_\_  
«\_\_» \_\_\_\_\_ 2024 року

Харків 2024

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 60 с., 2 табл., 20 рис., 4 дод., 31 джерело.

НЕЙРОННІ МЕРЕЖІ, ШТУЧНИЙ ІНТЕЛЕКТ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, РАДОНА, ПЕРЕТВОРЕННЯ ХАФА, ДЕТЕКТУВАННЯ ЛІНІЙ, ВІДРІЗКІВ, ІНТЕГРАЛЬНЕ ПЕРЕТВОРЕННЯ, ІНВАРІАНТНІ ПЕРЕТВОРЕННЯ, РОБАСТНІ ОЦІНЮВАЧІ.

Об'єктом дослідження є послідовність різноракурсних зображень об'єктів.

Метою дослідження є розробка методів, що базуються на використанні перетворення Радона, які дозволяють детектувати ознаки (відрізки ліній) та формувати вектори ознак для пошуку у базах даних.

Використано методи числового моделювання та аналітичного обґрунтування. Проведено дослідження методів пошуку зображення у базах даних та методів погодження різноракурсних зображень однієї сцени, детектування відрізків ліній на основі перетворень Радона і Хафа, та аналіз робастних методів детектування відрізків ліній. Досліджено метод детектування відрізків ліній за допомогою перетворення Радона, розроблено алгоритм детектування відрізків ліній.

У результаті дослідження здійснена програмна реалізація системи для пошуку в базах даних.

RADON TRANSFORM, TRANSFORM HOUGH, DETECTION LINE, SEGMENT, INTEGRAL TRANSFORMATION INVARIANT TRANSFORMATION, ROBUST APPRAISERS.

The object of the research is the sequence of multi-point image objects.

The aim of the research is to develop methods based on the use of Radon transforms, which allow to detect features (segments of lines) and to form feature vectors for search in databases.

Methods used numerical simulations and analytical reasoning. Research methods of finding images in databases and methods of coordination image of one scene detection line segments based on Hough and Radon transformation, and analysis of robust methods for detecting line segments. The method of detecting line segments using the Radon transform, the algorithm detecting line segments.

As a result of implemented software implementation of the system to search the database.

## ЗМІСТ

|  |           |
|--|-----------|
| Вступ.....   | 6         |
| 1 Аналіз методів класифікації зображень за допомогою нейромереж..... | 8         |
| 1.1 Поняття нейромережі .....  | 8         |
| 1.2 Згорткові нейронні мережі.....                                   | 10        |
| 1.2.1 Операція згортки .....   | 11        |
| 1.2.2 Пулінг .....   | 12        |
| 1.3 RNN .....  | 14        |
| 1.4 Класифікація зображень нейронними мережами .....                 | 16        |
| 1.5 Навчання нейронних мереж.....                                    | 17        |
| 1.5.1 Навчання з учителем.....                                       | 18        |
| 1.5.2 Навчання без учителя .....                                     | 19        |
| 1.6 Постановка задачі дослідження.....                               | 20        |
| 2 Методи класифікації зображень нейромережами.....                   | 21        |
| 2.1 Основні підходи класифікації зображень.....                      | 21        |
| 2.2 Попередня обробка даних .....                                    | 22        |
| 2.2.1 Локалізація зображення.....                                    | 23        |
| 2.2.2 Виявлення об'єктів .....                                       | 24        |
| 2.3 Архітектура згорткових нейронних мереж .....                     | 25        |
| 2.4 Сучасні архітектури.....   | 28        |
| 2.4.1 ResNet .....   | 28        |
| 2.4.2 EfficientNet .....   | 30        |
| 2.4.3 MobileNet .....  | 32        |
| 2.4.4 DenseNet .....   | 33        |
| 2.4.5 Vision Transformers .....                                      | 34        |
| 2.5 Метрики вимірювання точності моделі.....                         | 36        |
| <b>3 ДОСЛІДЖЕННЯ Комп'ютерної моделі фільтрації зображень .....</b>  | <b>39</b> |
| 3.1 Обґрунтування вибору середовища програмної реалізації .....      | 39        |
| 3.1.1 Jupyter Notebook .....   | 41        |

|   |    |
|---|----|
|   | 4  |
| 3.1.2 Бібліотеки для реалізації алгоритмів машинного навчання . | 43 |
| 3.2 Програмна реалізація.....                                   | 43 |
| 3.3 Інструкція користувача .....                                | 43 |
| 3.4 Тестування розробленої моделі.....                          | 43 |
| Висновки .....  | 44 |
| Перелік джерел посилання .....                                  | 45 |
| Додаток А.....  | 50 |

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

ІІІ – штучний інтелект

CV – Computer Vision (комп'ютерний зір)

CNN – Convolutional Neural Networks (згорткові нейронні мережі)

ViT – Vision Transformer (зоровий трансформер)

GPT – Generative Pre-trained Transformer (породжувальний попередньо тренований трансформер)

MLP – Multilayer Perceptron (багатошарового персептрон)

RNN – Recurrent Neural Networks (рекурентні нейронні мережі)

КТ – контрольні точки

YOLO – You Only Look Once (дивишся лише раз)

SSD – Single Shot MultiBox Detector (детектор розпізнавання за один прохід МультиБокс)

IDE – Integrated Development Environment (інтегроване середовище розробки)

## ВСТУП

Нейронні мережі є одним із напрямків машинного навчання, який виник ще в 1940-х роках. Цей підхід ґрунтується на біологічних нейронних мережах тварин, які складаються з нейронів головного та спинного мозку центральної нервової системи. Звісно, штучні нейронні мережі є значно спрощеною версією біологічних, і їхня мета полягає в розв'язанні завдань, які наш мозок виконує щодня.

В умовах стрімкого розвитку технологій, їх автоматизації, поширення робототехніки, розвитку інтернет-речей та систем ІІІ, одними з найбільш актуальних напрямів залишаються машинне навчання, комп'ютерний зір та розпізнавання об'єктів на зображеннях.

Нейронні мережі зробили революцію в області комп'ютерного зору, дозволивши машинам розпізнавати та аналізувати зображення. Вони стають все більш популярними завдяки своїй здатності вивчати складні моделі та особливості. Особливо CNN є найпопулярнішим типом нейронних мереж, які використовуються в обробці зображень. Але також ViT стають все більш популярними останнім часом завдяки революційним досягненням GPT та інших архітектур на основі трансформаторів у обробці природної мови. Загалом, нейронні мережі обробляють і розпізнають зображення різними способами. Це залежить від архітектури мережі та проблеми.

Нейронні мережі відіграють ключову роль у класифікації зображень, оскільки вони здатні автоматично навчатися розпізнавати візуальні патерни в даних. Використовуючи глибоке навчання, нейронні мережі можуть аналізувати величезні масиви зображень і класифікувати їх на основі різних ознак, таких як форми, кольори або текстури. Цей підхід дозволяє досягти високої точності в задачах класифікації зображень, що робить нейронні мережі незамінними для таких застосувань, як розпізнавання об'єктів, аналіз медичних зображень або автоматизація процесів в різних галузях.

Нейронні мережі використовують для задач, для яких побудувати алгоритм вирішення – складна та тривала для людини задача. Приклади таких задач:

- автоматизація контролю якості продукції;
- медична діагностика;
- ідентифікація підозрілих осіб у системах відеоспостереження;
- розпізнавання дорожніх знаків, пішоходів та інших автомобілів;
- класифікація зображень для автоматичного тегування фотографій, покращення рекомендацій та створення контенту в соціальних мережах;
- розпізнавання облич.

Актуальність дослідження полягає у зростаючій потребі в ефективних методах обробки та аналізу зображень у сучасному світі. У наш час обсяги даних, що генеруються щоденно, зростають із неймовірною швидкістю. Соціальні мережі, системи відеоспостереження, медичні пристрої – всі вони продукують величезні масиви зображень. Ефективні методи класифікації, зокрема ті, що ґрунтуються на нейронних мережах, дозволяють не лише обробляти ці дані, але й витягувати з них корисну інформацію.

# 1 АНАЛІЗ МЕТОДІВ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖ

## 1.1 Поняття нейромережі

Нейронна мережа – це програма або модель машинного навчання, яка приймає рішення подібно до людського мозку, використовуючи процеси, які імітують те, як біологічні нейрони працюють разом, щоб ідентифікувати явища, зважувати варіанти та робити висновки.

Кожна нейронна мережа складається з шарів вузлів або штучних нейронів – вхідного рівня, одного або кількох прихованих шарів і вихідного рівня. Нейронна мережа вважається неглибокою, якщо має лише один прихований шар (рис. 1.1), і глибокою, якщо кількість прихованих шарів більша за один (рис. 1.2). Шари містять в собі вузли, які є штучними нейронами. Кожен вузол підключається до інших і має власну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі.

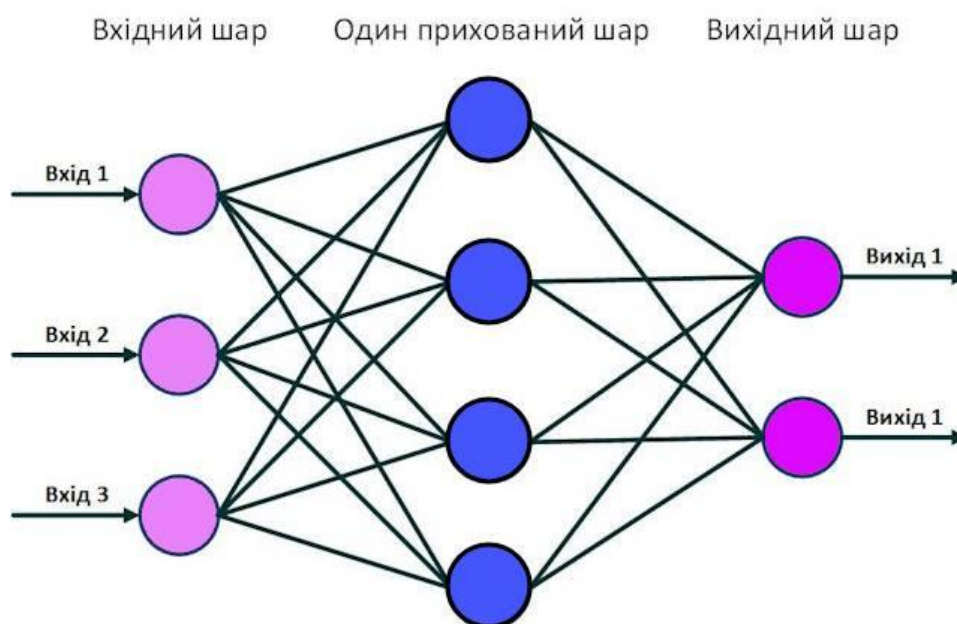


Рисунок 1.1 – Неглибока нейронна мережа



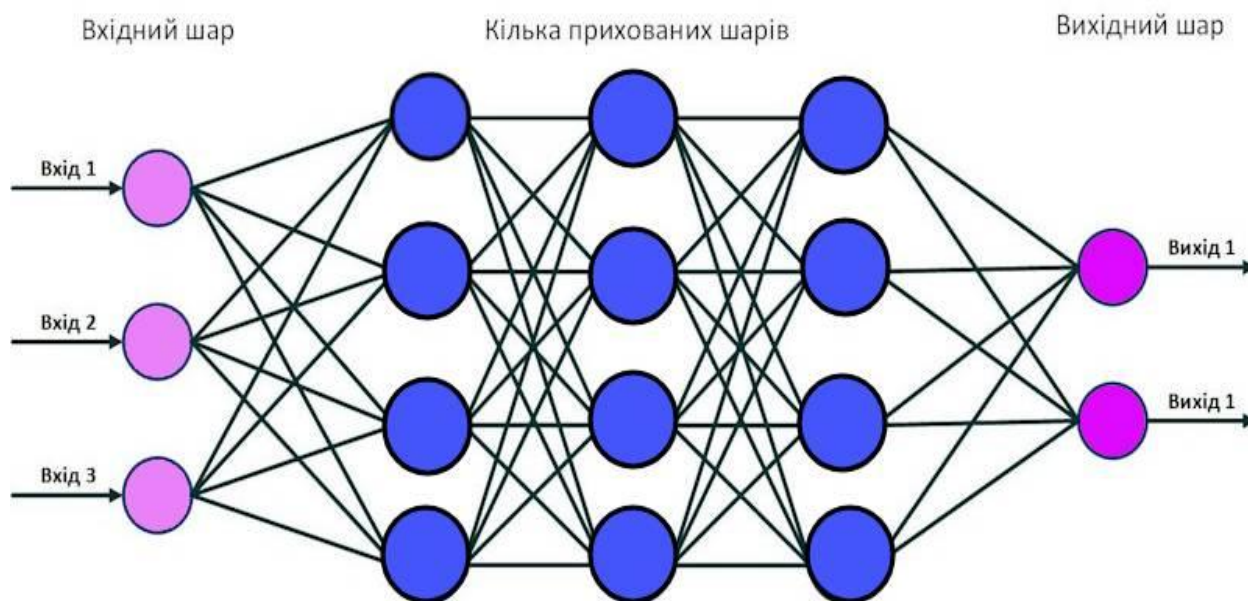


Рисунок 1.2 – Глибока нейронна мережа

Нейронні мережі покладаються на навчальні дані, щоб навчатися та підвищувати свою точність з часом. Після їх точного налаштування на точність вони стають потужними інструментами в інформатиці та штучному інтелекті, що дозволяють класифікувати та кластеризувати дані з високою швидкістю. Завдання з розпізнавання мовлення або розпізнавання зображень можуть тривати хвилини чи години порівняно з ручною ідентифікацією експертів-людей. Одним із найвідоміших прикладів нейронної мережі є пошуковий алгоритм Google.

Нейронні мережі іноді називають штучними нейронними мережами або імітованими нейронними мережами. Вони є підмножиною машинного навчання та є основою моделей глибокого навчання.

У сучасній практиці глибокі нейронні мережі часто використовують складніші архітектури, такі як CNN або RNN, які набагато краще підходять для складних задач. CNN спеціалізуються на обробці даних із просторовою структурою, таких як зображення або відео, де вони можуть автоматично виділяти просторові ознаки на різних рівнях. RNN, у свою чергу, відмінно працюють із послідовними даними, такими як текст, аудіо або часові ряди, завдяки здатності враховувати контекст попередніх елементів у послідовності.

Вони широко застосовуються в задачах обробки письмової та усної мови, генерації тексту, аналізу фінансових даних, розпізнавання мови та багатьох інших сферах, де важливий порядок даних або зв'язок між їх елементами.

## 1.2 Згорткові нейронні мережі

CNN є спеціальним типом нейронних мереж, який чудово підходить для роботи з зображеннями. Їх широко використовують у задачах комп'ютерного зору, таких як генерація й класифікація зображень, розпізнавання об'єктів і поз. Раніше ці проблеми намагалися вирішувати за допомогою класичних нейронних мереж, зокрема MLP (найпростіший варіант глибокої нейронної мережі), а також різними евристичними підходами, але CNN значно покращили результати в цій сфері.

Крім обробки двовимірних зображень, CNN також здатні працювати з одновимірними та тривимірними сигналами.

Наприклад, одновимірними даними можуть бути аудіосигнали, зібрані мікрофоном, які згорткові нейронні мережі використовують для задач розпізнавання мови (рис. 1.3).

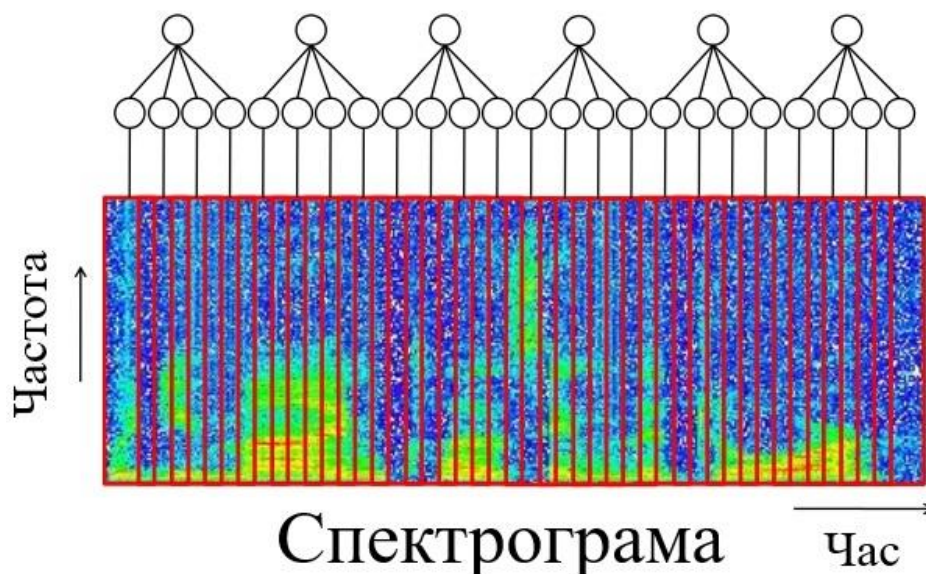


Рисунок 1.3 – Згорткова глибока нейронна мережа для звукового сигналу

Що ж до тривимірних даних, то ними є відео, де третім виміром є кількість кадрів в секунду, і CNN можуть аналізувати такі дані для розпізнавання об'єктів або дій у динаміці (рис. 1.4).

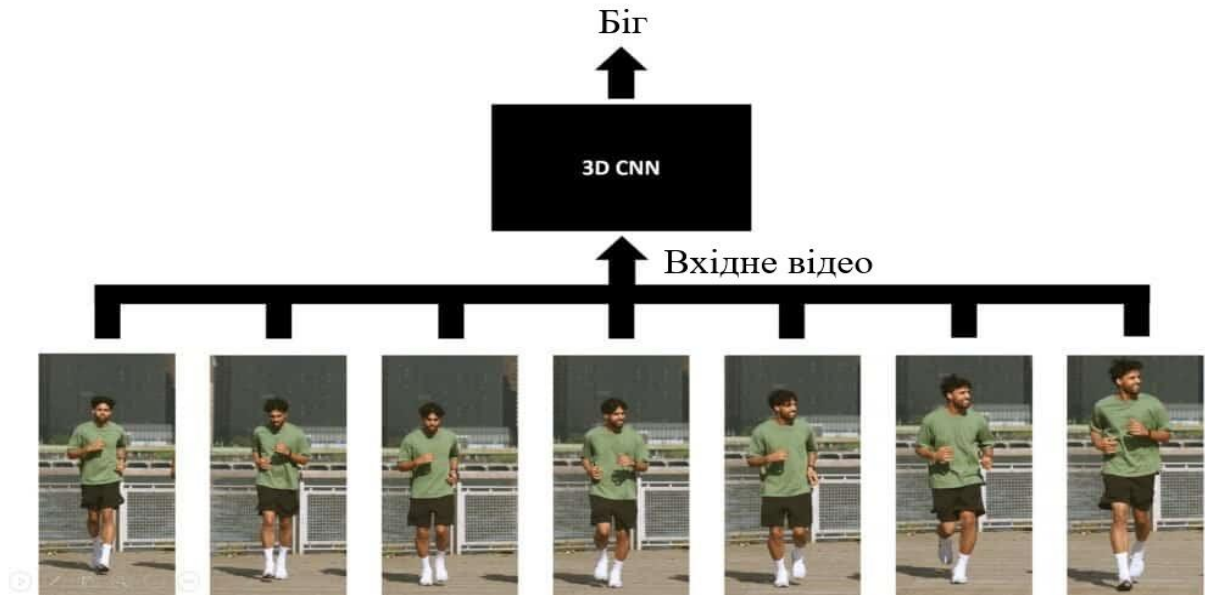


Рисунок 1.4 – Приклад розпізнавання бігу за допомогою CNN

### 1.2.1 Операція згортки

Згорткові нейронні мережі отримали свою назву через процес виявлення країв, де менші локальні ознаки зображення об'єднуються в більш загальні. Цей процес математично описується операцією, відомою як згортка, що виглядає наступним чином:

$$y[x, y] = f[x, y] * g[x, y] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[i, j] \cdot g[x - i, y - j], \quad (1.1)$$

де  $y[x, y]$  – значення вихідного сигналу;

$f[x, y]$  – вхідний сигнал;

$g[x, y]$  – ядро згортки, яке застосовується до вхідного сигналу.

Операція згортки в контексті обробки зображень включає поелементне множення частини вхідного зображення на ядро з подальшим сумуванням результатів. Це дає новий піксель у вихідному зображенні. Застосування згортки до зображення показано на рисунку 1.5.

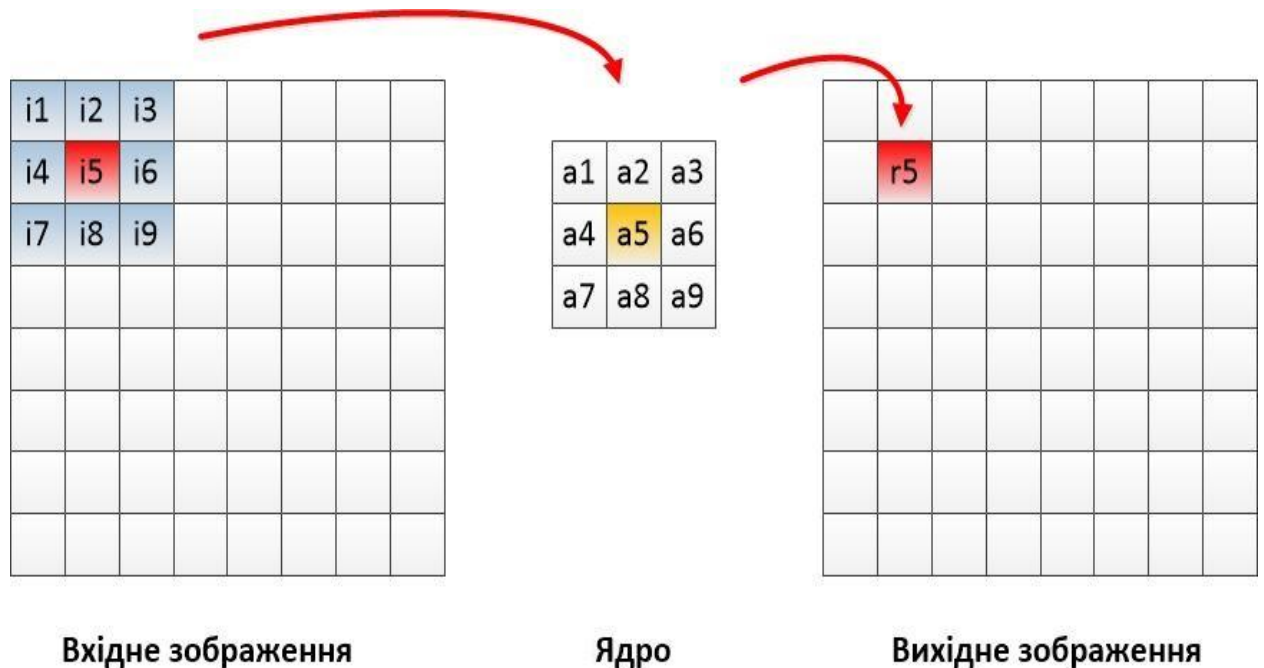


Рисунок 1.5 – Операція згортки зображення

### 1.2.2 Пулінг

Пулінг – це операція, яка використовується в CNN для зменшення розмірів вхідних даних, зберігаючи при цьому важливі характеристики. Це допомагає зменшити обчислювальну складність моделі та уникнути перенавчання.

Існує три основних види пулінгу:

– Max Pooling вибирає з кожного блоку вхідних даних максимальне значення. Ця процедура дозволяє зберігати найсильніші ознаки зображення. Наприклад, якщо є блок розміром  $2 \times 2$ , Max Pooling вибере найбільше значення в цьому блоці. Процедuru Max Pooling показано на рисунку 1.6;

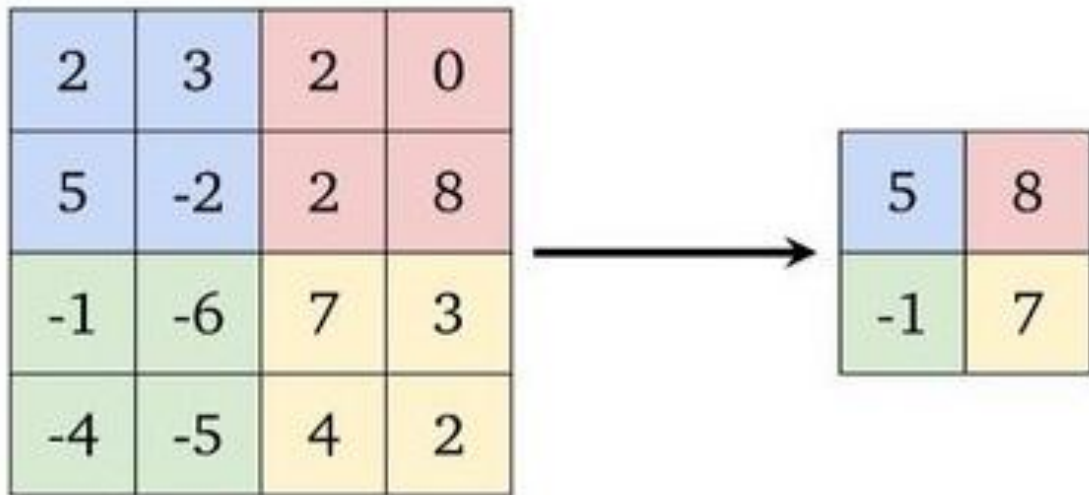


Рисунок 1.6 – принцип роботи Max Pooling

– Average Pooling обчислює з кожного блоку вхідних даних середнє значення. Цей підхід згладжує результати і може бути корисним у випадках, коли необхідно зберегти загальний контекст ознак. Результат роботи Average Pooling показано на рисунку 1.7;

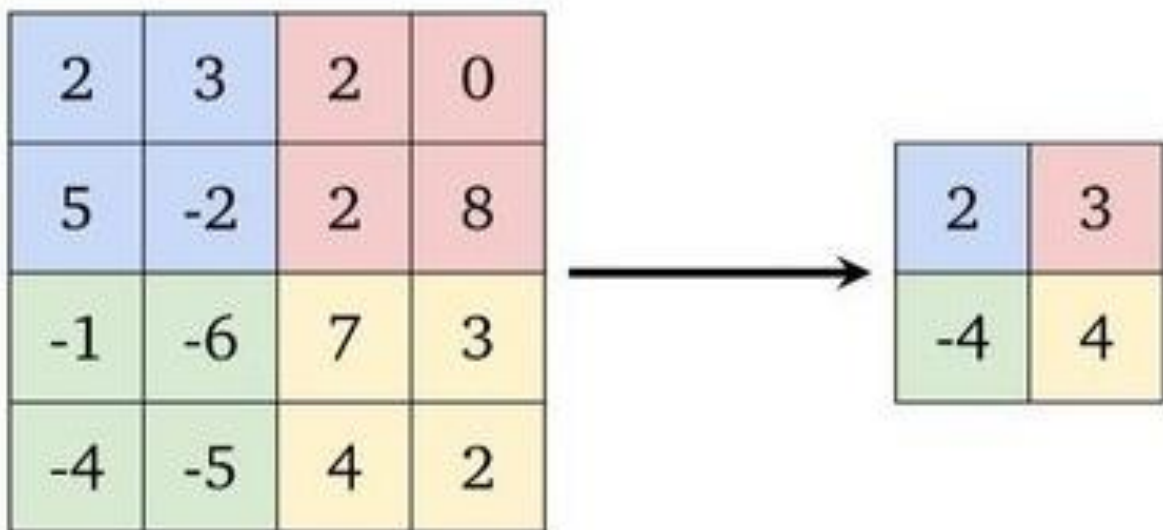


Рисунок 1.7 – принцип роботи Average Pooling

– Global Pooling зменшує всю карту ознак до одного значення на кожний канал. Наприклад, глобальний Max Pooling (рис. 1.8) або глобальний Average Pooling. Це часто використовується перед повнозв'язними шарами в CNN.



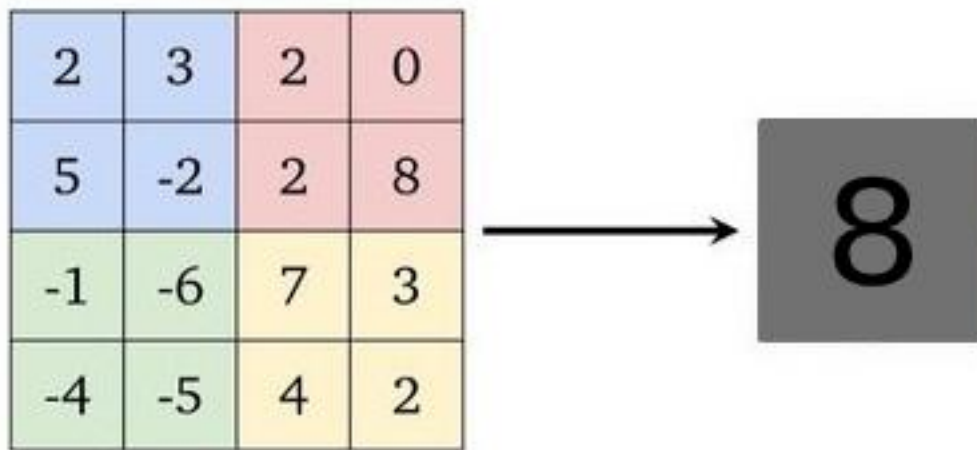


Рисунок 1.8 – приклад Global Max Pooling

Пулінг є критично важливою операцією в CNN. Основними перевагами пулінгу є те, що він дозволяє зменшити кількість параметрів і обчислень. Це зменшує обчислювальні витрати і потребу в пам'яті, прискорюючи роботу моделі. Також пулінг допомагає зробити мережу менш чутливою до дрібних зсувів або змін вхідних даних і скорочує кількість ознак, що знижує ризик перенавчання моделі. Усі ці переваги є важливими для завдань комп'ютерного зору, де об'єкти можуть з'являтися в різних положеннях або масштабах. Використання пулінгу, зокрема максимального, також забезпечує збереження важливих інформаційних елементів, що може покращити точність класифікації в задачах комп'ютерного зору.

### 1.3 RNN

RNN зазвичай використовуються для обробки послідовних даних, таких як текст або часові ряди, їх також можна застосувати для класифікації зображень, при умові якщо розглядати зображення як послідовність пікселів або блоків пікселів. На відміну від CNN, RNN здатні запам'ятовувати ключові деталі про отримані вхідні дані завдяки своїй внутрішній пам'яті (рис. 1.9), що дозволяє їм з високою точністю передбачати майбутні події.

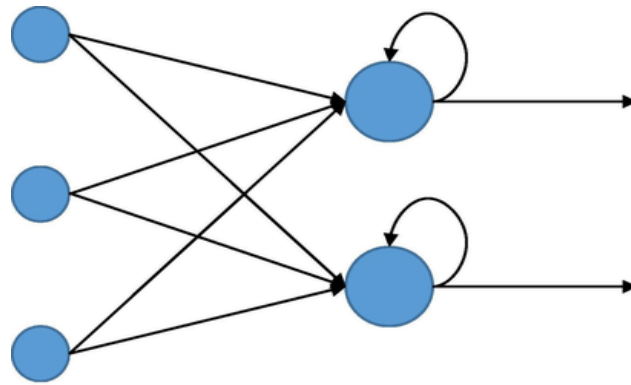


Рисунок 1.9 – рекурентна нейромережа

RNN не є стандартним вибором для класифікації зображень, вони можуть бути корисними в спеціальних випадках або в поєднанні з CNN. Найчастіше використання RNN виправдано, коли потрібно враховувати послідовну структуру у зображеннях, наприклад, у задачах розпізнавання рукописного тексту або класифікації зображень з певною послідовністю елементів.

Основна ідея полягає в тому, щоб розглядати зображення як двовимірну сітку пікселів і перетворювати її в послідовність. Наприклад, можна «прочитати» зображення піксель за пікселем, рядок за рядком або блок за блоком. Кожен рядок або блок пікселів можна розглядати як елемент послідовності, який передається на вхід RNN.

Кожен крок RNN обробляє черговий елемент послідовності і передає стан на наступний крок, поступово «запам'ятовуючи» контекст зображення. Після обробки всієї послідовності фінальний стан RNN можна використовувати для класифікації зображення.

Найбільш ефективним підходом є поєднання CNN з RNN. CNN спочатку використовується для вилучення просторових ознак зображення, обробляючи локальні патерни за допомогою згорток. Це допомагає зберегти ключову інформацію про форму, текстуру та інші характеристики зображення. Потім ці ознаки можуть подаватися в RNN для обробки вже як послідовності високорівневих ознак. Таким чином, RNN допомагає враховувати глобальний контекст або залежності між різними частинами зображення.

## 1.4 Класифікація зображень нейронними мережами

За останні роки задача класифікації зображень стала дуже популярною завдяки зростанню кількості проєктів, спрямованих на автоматизацію керування транспортними засобами, розпізнавання облич, автомобільних номерів та інших об'єктів за допомогою камер. Такі технології активно застосовуються у різних сферах нашого життя. Класифікація полягає в ідентифікації об'єкта на зображенні та віднесенні його до певної категорії (рис. 1.10).

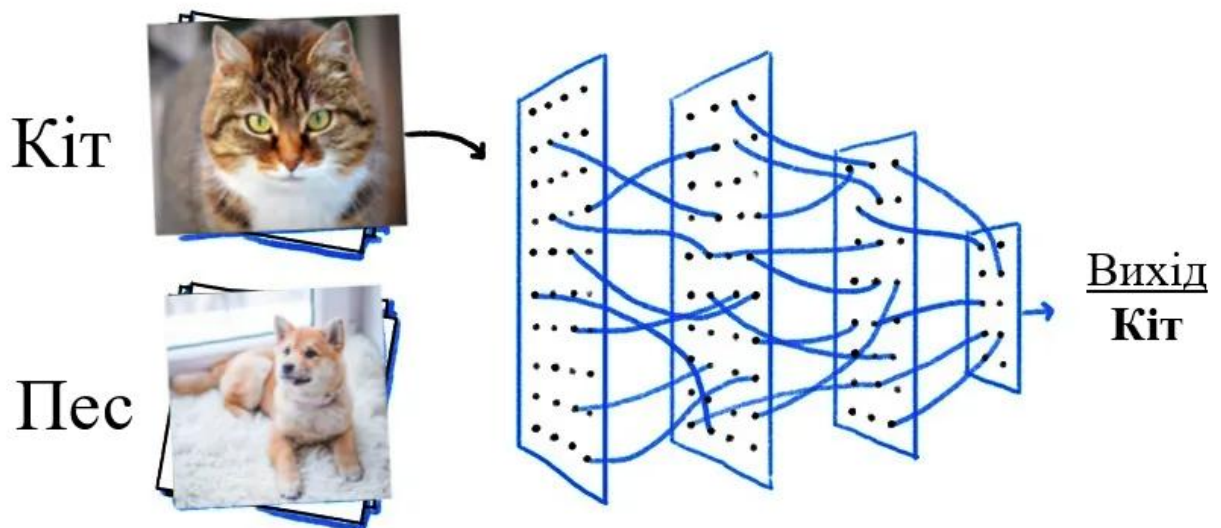


Рисунок 1.10 – Приклад класифікації зображень

Алгоритми машинного навчання, що використовуються для класифікації зображень, постійно вдосконалюються експертами, однак загальна структура їх виконання залишається незмінною. Процес класифікації зазвичай виконується покроково:

Крок 1. Нормалізація даних. Це початковий етап, де зображення готують до подальшої обробки. Включає, наприклад, масштабування або зменшення шуму, щоб забезпечити рівні умови для роботи алгоритму;



Крок 2. Виділення та сегментація об'єкта. На цьому етапі на зображенні виділяють ту частину, де знаходиться об'єкт, який необхідно класифікувати. Це може бути як простий контур, так і складна форма;

Крок 3. Виділення ключових ознак. Далі визначаються найважливіші характеристики об'єкта, такі як форма, текстура, колір тощо. Такі ознаки дозволяють алгоритму «зрозуміти», з чим він має справу;

Крок 4. Класифікація. Нарешті, на основі виділених ознак, об'єкт порівнюється з відомими класами, і йому присвоюється певна категорія.

Цей базовий процес може мати додаткові кроки, такі як аугментація даних – це штучне збільшення кількості навчальних зразків для підвищення точності моделі. Також, якщо використовуються сучасні глибокі нейронні мережі, вони можуть автоматично виділяти важливі ознаки, що значно спрощує процес і підвищує точність при роботі зі складними зображеннями.

## 1.5 Навчання нейронних мереж

У ході навчання нейронної мережі модель штучного інтелекту вчиться виконувати певні завдання з урахуванням наданих їй даних.

Головне завдання навчання нейромереж полягає у розвитку у них здатності вирішувати поставлені завдання. Деякими з цих завдань можуть бути:

- автоматизація процесів;
- прогнозування;
- обробка великих масивів даних;
- покращення якості прийняття рішень;
- регресія;
- кластеризація.

Є безліч методів «тренування» нейронних мереж, але вони ґрунтуються на двох ключових принципах: за допомогою вчителя і без нього. Це

відбувається так само, як і в людини: можна набувати нових знань під керівництвом наставника, який підкаже та скоригує окремі моменти, а можна займатися самоосвітою. В останньому випадку людина спирається лише на свій особистий досвід та спостереження.

### 1.5.1 Навчання з учителем

Навчання з учителем – це найбільш поширений підхід, при якому нейромережа навчається на основі попередньо розмічених даних. Ці дані включають вхідні значення і відповідні їм цільові. Нейронна мережа навчається передбачати останні на основі вхідних даних.

Процес навчання відбувається наступним чином. Спочатку потрібно підготувати набір вхідних даних, для яких відомі очікувані результати. Наприклад, якщо ми хочемо навчити нейронну мережу розпізнавати зображення кішок, нам потрібно підготувати набір зображень кішок (вхідні дані) та міток, які вказують, що на цих зображеннях дійсно зображені кішки (очікувані результати).

На етапі процесу навчання нейронна мережа переглядає вхідні дані та робить передбачення на основі своїх поточних параметрів (ваг). Потім ці прогнози порівнюються з очікуваними результатами.

Якщо прогноз нейронної мережі не відповідає очікуваному результату, запускається процес, відомий як зворотне поширення помилки. У цьому процесі визначається, як кожна вага в нейромережі вплинула на загальну помилку. Потім ці ваги коригуються в напрямку, що дозволяє зменшити цю помилку.

Цей процес повторюється багато разів, використовуючи нові дані з навчального набору. З часом ваги нейронної мережі налаштовуються так, щоб мінімізувати розбіжність між прогнозами та очікуваними результатами.

Далі навчені нейронні мережі перевіряються на нових даних, які раніше не використовувалися в процесі навчання. Це дає змогу оцінити, наскільки ефективно модель може застосовувати свої знання в нових ситуаціях.

Важливо, що навчання з викладачем вимагає великої кількості розмічених даних, що може бути дорогим та трудомістким процесом. Для досягнення хороших результатів навчання важливо мати якісні та точні мітки. Неправильні або неякісні дані можуть призвести до навчання моделі з помилками, що знижує її ефективність. Забезпечення високої якості даних вимагає додаткових ресурсів і зусиль.

### 1.5.2 Навчання без учителя

Навчання без учителя – це процес, під час якого нейронна мережа оперує нерозміченими даними. Головна мета полягає у виявленні прихованих закономірностей, структур або взаємозв'язків у даних без будь-якої попередньої інформації про очікувані результати.

Процес починається з підготовки даних. На відміну від навчання з учителем, де дані повинні бути міченими, тут розмітка не потрібна. Достатньо мати лише набір вхідних показників. Наприклад, для навчання нейронної мережі групувати зображення котів і собак достатньо просто зібрати їх зображення, не вказуючи, де коти, а де собаки.

Після цього нейронна мережа приступає до навчання, намагаючись виявити структуру або закономірності в цих даних. Це можуть бути процеси кластеризації, коли подібні об'єкти групуються разом, або пошук аномалій, де виділяються об'єкти, що суттєво відрізняються від інших. Також можливо виконання зменшення розмірності – скорочення кількості ознак без суттєвої втрати інформації.

Протягом навчання ваги нейронної мережі постійно коригуються, щоб точніше відображати структуру даних. Процес проходить кілька ітерацій до

того моменту, поки мережа не налаштується на оптимальне відображення знайдених закономірностей. Завершальним етапом є тестування навченої моделі на нових, раніше невикористаних даних для оцінки того, наскільки ефективно вона виявляє структури в нових ситуаціях.

## 1.6 Постановка задачі дослідження

Методи та алгоритми класифікації зображень залишаються актуальними, оскільки допомагають вирішувати реальні задачі в різних сферах, де активно використовуються зображення – таких як медицина, технології, мистецтво, наука та інші. Спираючись на це, важливо дослідити застосування нейронних мереж для класифікації зображень з урахуванням можливостей сучасних технологій.

Об'єктом дослідження є метод класифікації зображень за допомогою нейронних мереж, зокрема його застосування в сучасних технологічних умовах.

Метою дослідження є розроблення, реалізація та дослідження методу, що базується на використанні нейронних мереж для класифікації зображень, які здатні ефективно виділяти ознаки.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів класифікації зображень за допомогою нейронних мереж;
- дослідити підходи для виділення ключових ознак зображень за допомогою нейромереж;
- розробити метод класифікації зображень на основі нейронних мереж із застосуванням сучасних методів обробки даних;
- створити комп'ютерну модель для автоматизованого процесу класифікації зображень та аналізу результатів;
- провести порівняння запропонованого методу з існуючими підходами.

## 2 МЕТОДИ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ НЕЙРОМЕРЕЖАМИ

### 2.1 Основні підходи класифікації зображень

Існує два основних підходи класифікації зображень: класифікація на основі пікселів та класифікація на основі об'єктів.

Пікселі є базовими елементами зображення і класифікація на основі аналізу пікселів є основним методом. Існують алгоритми, що або використовують тільки спектральну інформацію окремих пікселів для класифікації, або поєднують спектральну інформацію з просторовою (дані про прилеглі пікселі). Методи класифікації на рівні пікселів враховують лише яскравість кожного пікселя, тоді як об'єктна класифікація використовує як спектральну, так і просторову інформацію.

До методів класифікації на основі пікселів належать такі, як метод мінімальної відстані до середнього, максимальна ймовірність та метод відстані Махаланобіса. Вони використовують середні значення та дисперсії для різних класів і працюють за принципом вимірювання «відстані» між цими середніми значеннями та значеннями пікселів. Методи класифікації на основі пікселів мають обмеження, оскільки не враховують інформацію з сусідніх пікселів, але мають перевагу в простоті реалізації та ефективності у задачах з базовими візуальними характеристиками.

Натомість класифікація на рівні об'єктів враховує і спектральні, і просторові дані, оскільки включає прилеглі пікселі в процес класифікації. Під «об'єктом» розуміють не конкретний об'єкт на зображенні, а суміжну область пікселів. Цей підхід базується на розпізнаванні об'єктів або груп пікселів, які утворюють певні смислові одиниці на зображенні. Він дозволяє враховувати не тільки самі пікселі, але й їх розташування та взаємозв'язки між ними.

Таким чином, класифікація на основі пікселів більше підходить для простих задач, тоді як класифікація на основі об'єктів є більш універсальною та потужною для складних візуальних задач.

## 2.2 Попередня обробка даних

Сучасні та надійні системи класифікації зображень здебільшого використовують підхід, орієнтований на об'єкти, де зображення потрібно відповідним чином підготувати. Спочатку необхідно вибрати та попередньо обробити об'єкти або області на зображенні.

Перед тим як класифікувати зображення чи його окремі об'єкти, комп'ютер повинен «зрозуміти» дані, які воно містить. Це досягається шляхом попередньої обробки та підготовки зображень для подальшого введення в класифікаційний алгоритм, використовуючи техніки виявлення об'єктів. Цей процес є важливим етапом підготовки зображень і даних для навчання моделей машинного навчання.

Попередня обробка зображень є важливою для застосунків комп'ютерного зору, оскільки вона покращує точність до 30% [6]. Наступні етапи попередньої обробки відіграють дуже важливу роль в оптимізації зображень для аналізу, оскільки покращують вхідні дані для моделей машинного навчання:

- зменшення шуму;
- підвищення контрастності;
- зміна розміру;
- нормалізація;
- аугментація;
- перетворення кольорових просторів (наприклад, з RGB у відтінки сірого або HSV);
- виділення контурів;
- підвищення різкості;
- видалення небажаних артефактів.

Правильна попередня обробка зображень не тільки підвищує точність, але й знижує обчислювальну складність, полегшуючи моделі навчання важливим ознакам і покращуючи їх роботу в реальних додатках.

### 2.2.1 Локалізація зображення

Виявлення об'єктів здійснюється за допомогою різних методів. Спосіб попередньої обробки залежить від того, чи на зображенні присутній один об'єкт інтересу, чи їх кілька. Якщо на зображенні є лише один об'єкт, застосовується техніка локалізації зображення. Пікселі зображення мають числові значення, що інтерпретуються комп'ютером для відтворення відповідних кольорів і відтінків. Навколо об'єкта, що цікавить, створюється обмежувальна рамка, яка вказує комп'ютеру, яка частина зображення є важливою та які піксельні значення її описують.

Локалізація є важливим етапом у процесі розпізнавання об'єктів, оскільки вона допомагає не тільки визначити наявність об'єкта, але й точно вказати його місцезнаходження на зображенні. Обмежувальна рамка дозволяє моделі фокусуватися на конкретній частині зображення, що спрощує подальшу класифікацію (рис. 2.1). У багатьох випадках локалізація може бути доповнена додатковими параметрами, наприклад, оцінкою впевненості в тому, що об'єкт дійсно є в межах обмежувальної рамки. Ця інформація особливо важлива в складних або завантажених сценах, де можуть бути кілька схожих об'єктів або де частини об'єктів можуть бути частково закриті іншими елементами зображення.



Рисунок 2.1 – локалізація зображення

### 2.2.2 Виявлення об'єктів

Якщо ж на зображенні декілька об'єктів, застосовується метод виявлення об'єктів (рис. 2.2), щоб визначити обмежувальні рамки для кожного об'єкта. Цей метод дозволяє одночасно ідентифікувати та локалізувати кілька об'єктів на одному зображенні, що є важливим аспектом у багатьох застосуваннях комп'ютерного зору. Виявлення об'єктів може виконуватися за допомогою різних алгоритмів, таких як YOLO, SSD та Faster R-CNN. Кожен з цих алгоритмів має свої переваги та недоліки, проте всі вони намагаються оптимізувати процес виявлення, щоб зменшити час обробки та підвищити точність. Наприклад, YOLO обробляє зображення за один прохід, що дозволяє досягти високої швидкості в реальному часі, тоді як Faster R-CNN використовує попередньо обчислені регіональні пропозиції, що забезпечує високу точність, але вимагає більше часу на обробку. Після виявлення об'єктів, для кожного з них створюються обмежувальні рамки, які вказують на їх точне розташування.

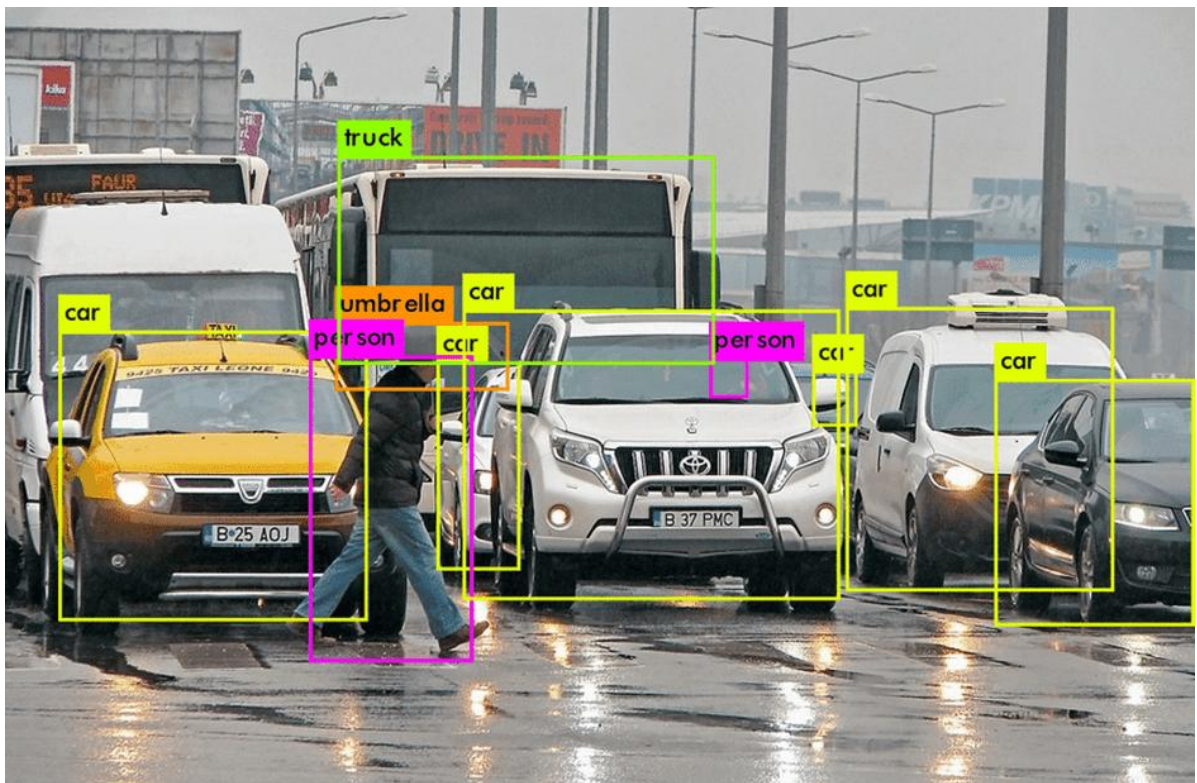


Рисунок 2.2 – виявлення об'єктів на зображенні



### 2.3 Архітектура згорткових нейронних мереж

Класична архітектура CNN складається з різних шарів, кожен з яких виконує унікальні функції, необхідні для ефективного аналізу зображень. Ці шари взаємодіють для вилучення ознак, їх обробки і, зрештою, класифікації. Основні типи шарів у CNN:

- згортковий шар. Цей шар є основою нейронної мережі. У цьому шарі використовуються невеликі фільтри для сканування зображення і виявлення різних ознак, таких як краї, кути або прості геометричні фігури. Кожен фільтр згортається по зображенню, виконуючи операцію згортки, щоб створити карти ознак, які містять інформацію про специфічні патерни в зображенні. Це дозволяє мережі «бачити» локальні структури на зображенні і поступово виділяти більш складні ознаки на глибших рівнях мережі;

- пулінговий шар використовується для зменшення розмірності карт ознак, що отримані після згорткових шарів. Це зменшує кількість параметрів і обчислень у мережі, зберігаючи найбільш важливі ознаки;

- повнозв'язні шари розташовані в кінці архітектури нейронної мережі і відповідають за остаточну класифікацію. У повнозв'язних шарах кожен нейрон пов'язаний з усіма нейронами попереднього шару, що дозволяє мережі об'єднати всю вилучену інформацію з попередніх згорткових і пулінгових шарів. Основна функція цього шару – узагальнення виділених ознак і їх перетворення в ймовірності для кожного можливого класу. На виході останнього повнозв'язного шару зазвичай використовується функція Softmax, яка перетворює значення у ймовірності класів, забезпечуючи вибір найбільш ймовірного класу для зображення.

Класична архітектура CNN (рис. 2.3) складається з компонентів, що працюють разом для ефективного аналізу зображень. Згорткові шари виділяють ознаки, пулінгові шари зменшують розмірність даних, а повнозв'язні шари виконують остаточну класифікацію. Разом ці компоненти створюють потужну систему для вирішення завдань комп'ютерного зору.

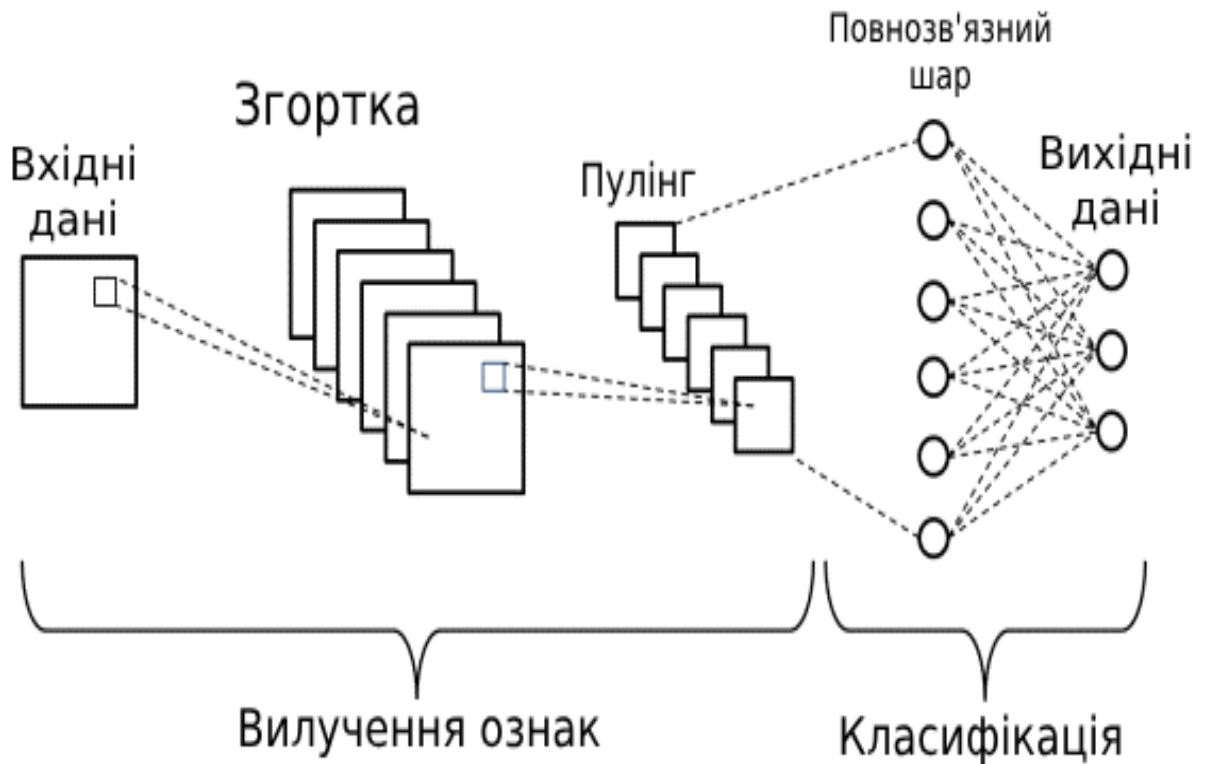


Рисунок 2.3 – класична архітектура CNN

Процедура класифікації за допомогою CNN здійснюється за наступною схемою:

Крок 1. На вхід мережі подається зображення, яке представляється як тривимірний тензор (ширина, висота, кількість каналів). Наприклад, для кольорового зображення розміром  $32 \times 32$  пікселів, вхід буде  $32 \times 32 \times 3$  (3 канали для кольорів RGB).

Крок 2. До вхідного зображення застосовується згортковий шар. У цьому шарі використовуються кілька фільтрів (наприклад,  $5 \times 5$ ), які сканують зображення і виділяють локальні ознаки, такі як краї або текстури.

Крок 3. Кожен фільтр створює карту ознак, яка показує, де на зображенні присутні виділені фільтром ознаки. Зазвичай кілька фільтрів використовуються одночасно, що дозволяє отримати кілька карт ознак.

Крок 4. Застосовується пулінговий шар, зазвичай максимальний пулінг. Він зменшує розмір карт ознак, зберігаючи найважливішу інформацію. Наприклад, максимальний пулінг з вікном  $2 \times 2$  вибирає максимальне значення

в кожній області розміру  $2 \times 2$  і створює зменшену карту ознак. Це зменшення розмірності допомагає знизити кількість параметрів і обчислень, підвищуючи ефективність мережі.

Крок 5. Згорткові та пулінгові шари зазвичай повторюються кілька разів для виділення більш складних і абстрактних ознак на різних рівнях. На кожному етапі глибина мережі збільшується, і мережа навчається розпізнавати складніші патерни.

Крок 6. Після кількох рівнів згорткових і пулінгових шарів, тривимірні карти ознак перетворюються у вектор (одномірний масив) за допомогою процесу, званого флатенінг. Це потрібно для підготовки даних до повнозв'язних шарів.

Крок 7. Вектор передається на вхід повнозв'язного шару, де кожен нейрон з'єднаний з усіма нейронами попереднього шару. Це дозволяє об'єднати інформацію з усіх ознак, виділених у попередніх шарах.

Крок 8. Повнозв'язні шари працюють на рівні класифікації і допомагають мережі передбачати належність зображення до одного з класів.

Крок 9. Останній шар мережі – це вихідний шар, який використовує функцію Softmax для багатокласової класифікації. Ця функція перетворює значення на виході повнозв'язного шару в ймовірності для кожного класу.

Крок 10. На основі ймовірностей мережа робить остаточне передбачення, визначаючи клас, до якого належить зображення.

Крок 11. Під час навчання мережа проходить через цикли прямого і зворотного поширення помилки. У прямому проході мережа обчислює передбачення, а в зворотному — коригує свої ваги, щоб зменшити різницю між передбаченими значеннями і реальними мітками зображень.

Крок 12. Після навчання мережа оцінюється на тестовій вибірці, щоб перевірити її здатність класифікувати нові, невідомі раніше зображення. Якщо точність передбачень недостатня, архітектуру мережі можна вдосконалити, додавши або змінюючи кількість згорткових шарів, використовуючи регуляризацию або інші методи.

## 2.4 Сучасні архітектури

Глибоке навчання стало ключовим підходом у класифікації зображень завдяки здатності CNN автоматично вивчати високорівневі ознаки зображень. Із розвитком обчислювальних потужностей та збільшенням кількості даних дослідники почали створювати більш складні архітектури CNN, які дозволяють покращити точність моделей та їх ефективність.

### 2.4.1 ResNet

ResNet (Residual Network) є однією з найвідоміших і найуспішніших архітектур у галузі глибокого навчання, яка зробила прорив у тренуванні дуже глибоких нейронних мереж. Вона була запропонована в 2015 році командою дослідників з Microsoft. Основною ідеєю ResNet є використання залишкових з'єднань, що дозволяють обійти проблеми тренування глибоких мереж, зокрема зникання градієнта.

У класичних глибоких нейронних мережах із великою кількістю шарів виникає проблема деградації градієнта під час навчання – коли мережа стає дуже глибокою, тренування може ставати менш ефективним, а точність навіть зменшується. Це відбувається через те, що градієнт, який передається назад під час зворотного поширення, може зникати або занадто зменшуватися з кожним шаром, що заважає ефективному оновленню ваг.

У класичних глибоких мережах збільшення кількості шарів може призвести до зниження точності через труднощі з оптимізацією, зокрема через проблему зникання або вибуху градієнтів. ResNet вирішує цю проблему за допомогою залишкових з'єднань (рис. 2.4), додаючи прямі шляхи передачі інформації від одного шару до іншого, оминаючи деякі проміжні шари. Саме це дозволяє шарам краще зберігати корисну інформацію та полегшує тренування.

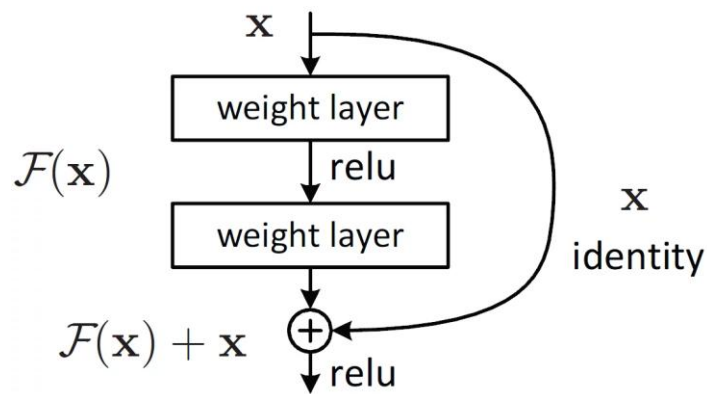


Рисунок 2.4 – залишкове з'єднання

ResNet має кілька варіацій, які відрізняються кількістю шарів: ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152. Число в назві вказує на кількість шарів у мережі, причому ResNet-18 є найменш глибокою архітектурою, а ResNet-152 – найглибшою. Приклад ResNet-50 зображено на рисунку 2.5.

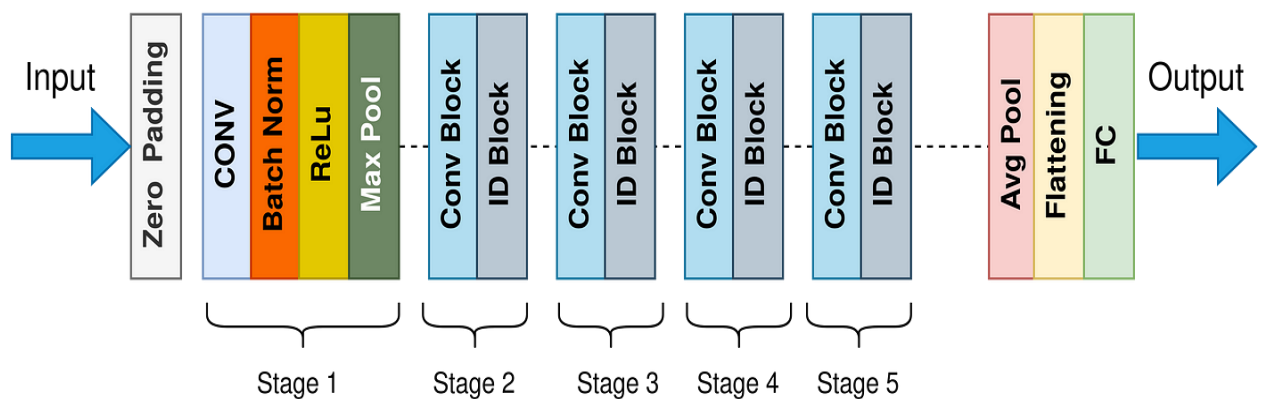


Рисунок 2.5 – архітектура 50-шарової моделі ResNet

З переваг ResNet забезпечує високу точність класифікації, завдяки чому є одним із провідних рішень для задач комп'ютерного зору. Архітектура дозволяє ефективно навчати дуже глибокі моделі, завдяки чому можливе тренування нейронних мереж із сотнями шарів без втрати продуктивності. Також ResNet відмінно справляється з обробкою великих наборів даних, що робить її надійною для складних завдань. Гнучкість і модульність архітектури дає змогу легко адаптувати її для різних варіантів завдань.

### 2.4.2 EfficientNet

EfficientNet – це сучасна архітектура нейронної мережі, яка була представлена компанією Google у 2019 році. Її основною метою є забезпечення максимальної продуктивності і ефективності обчислень, досягаючи високої точності при меншому споживанні ресурсів порівняно з іншими великими моделями для класифікації зображень.

EfficientNet-B0 – це базова версія мережі. Після базової моделі були розроблені масштабовані версії від EfficientNet-B1 до EfficientNet-B7. Кожна з цих моделей має більше шарів і фільтрів, а також обробляє зображення з більш високою роздільною здатністю, що дозволяє досягти кращих результатів на великих наборах даних. Різницю майбутніх поколінь EfficientNet у розмірах вхідних зображень показано у таблиці 2.1.

Таблиця 2.1 – різниця роздільної здатності EfficientNet

| Ім'я моделі     | Вхідне зображення |
|-----------------|-------------------|
| EfficientNet-B0 | 224×224           |
| EfficientNet-B1 | 240×240           |
| EfficientNet-B2 | 260×260           |
| EfficientNet-B3 | 300×300           |
| EfficientNet-B4 | 380×380           |
| EfficientNet-B5 | 456×456           |
| EfficientNet-B6 | 528×528           |
| EfficientNet-B7 | 600×600           |

Основним будівельними блоками EfficientNet є інвертовані «вузькі» шари (MBConv), що показано на рисунку 2.6. Вони оптимізовані для зменшення обчислювальних витрат і використання пам'яті. Ці блоки дозволяють мережі зберігати важливі особливості зображень, водночас зменшуючи кількість обчислень.

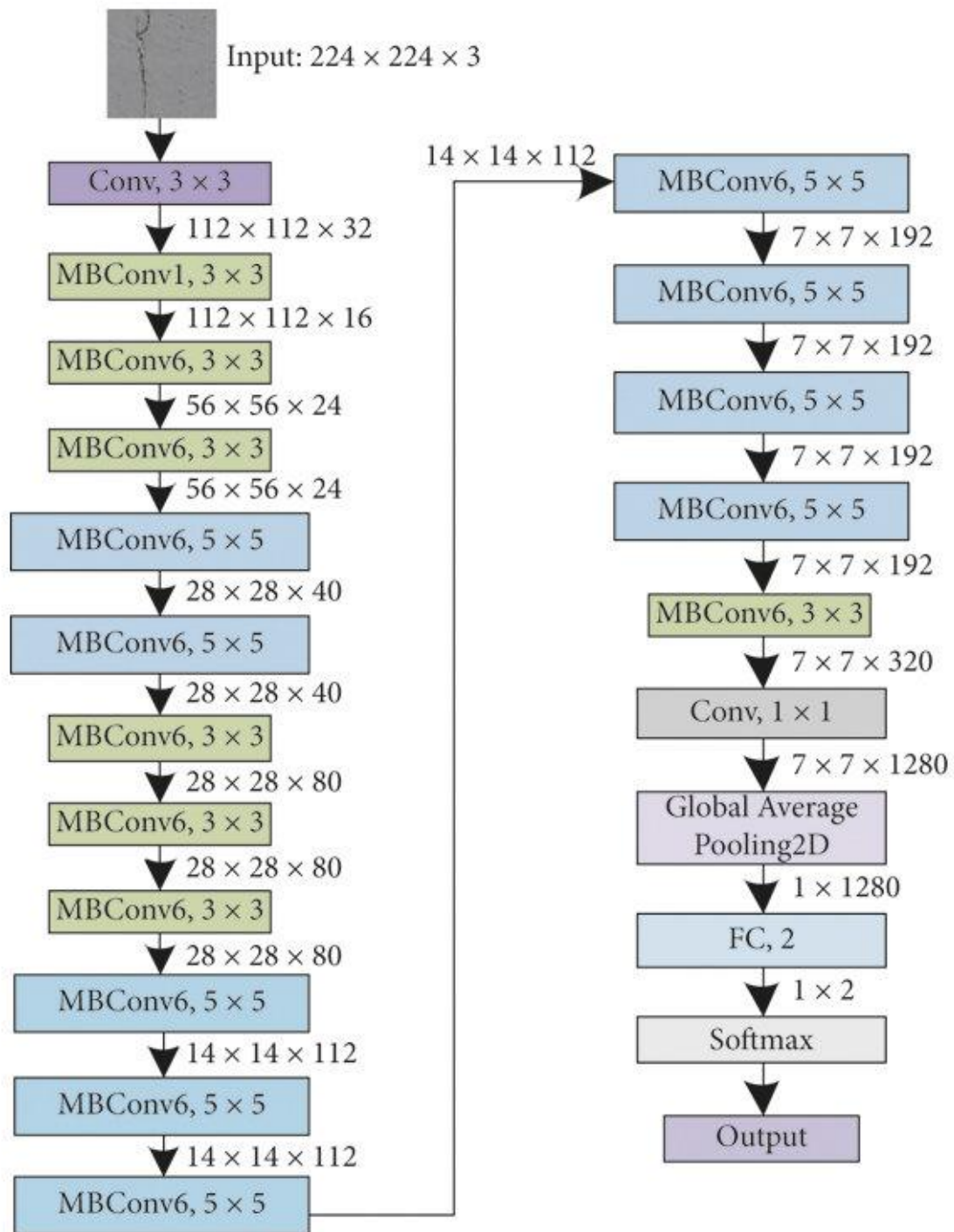


Рисунок 2.6 – архітектура EfficientNet-B0 з використанням MBConv

EfficientNet демонструє значну перевагу в порівнянні з іншими великими архітектурами. Наприклад, EfficientNet-B7 досягає точності понад 84% на ImageNet, при цьому використовуючи на порядок менше параметрів і обчислень. Це робить модель дуже привабливою для додатків із обмеженими ресурсами, таких як мобільні пристрої та системи вбудованого типу.

### 2.4.3 MobileNet

MobileNet розроблена компанією Google для ефективної класифікації зображень і інших задач комп'ютерного зору для використання в мобільних і вбудованих системах з обмеженими ресурсами. Вона застосовує розподілені згорткові операції, що суттєво знижує кількість параметрів і обчислювальних операцій, зберігаючи прийнятний рівень точності.

Оскільки MobileNet використовує згортки, розділені на два етапи – глибинну згортку (обробка кожного каналу окремо) і точкову згортку (об'єднання каналів), це призводить до значного зниження обчислювальної складності. MobileNet дозволяє регулювати кількість каналів і роздільну здатність мережі, що дає можливість збалансувати точність моделі та ефективність обчислень.

Архітектура MobileNet (рис. 2.7) має на даний момент 3 версії. Перша версія оптимізована для мобільних пристроїв завдяки використанню глибинно-згорткових шарів. Друга версія додає шар інверсійного залишку і лінійні шари згортки, що допомагають зменшити втрати інформації. У третій версії застосовано пошук архітектури нейронної мережі для оптимізації параметрів, а також додано блоки з активацією Swish, що додатково підвищує ефективність роботи моделі.

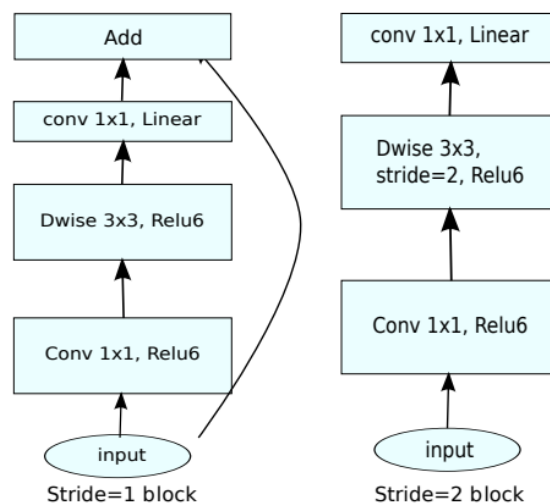


Рисунок 2.7 – приклад архітектури MobileNetV2



Завдяки своїй ефективності, MobileNet добре підходить для реального часу та використання на пристроях з обмеженими ресурсами, де важливо мати високу продуктивність і невеликий розмір моделі.

#### 2.4.4 DenseNet

Ідею щільних з'єднань між шарами пропонує DenseNet, де кожен шар отримує вхід з усіх попередніх шарів і передає свої виходи всім наступним. Це означає, що всі шари тісно з'єднані, і кожен шар отримує інформацію від усіх попередніх шарів, що сприяє зменшенню проблеми зникнення градієнта, полегшує навчання та підвищує ефективність використання параметрів мережі.

Архітектура DenseNet складається з перехідних шарів і щільних блоків. Кожен згортковий шар усередині щільного блоку пов'язаний з кожним іншим шаром у цьому блоці. Це досягається шляхом підключення виходу кожного шару до входу наступного шару, що створює «коротке» з'єднання. Перехідні шари зменшують розмір карт ознак між щільними блоками, що дозволяє мережі ефективно розширюватися. Як і в інших нейромережах, у DenseNet існує кілька версій, які різняться глибиною та кількістю параметрів.

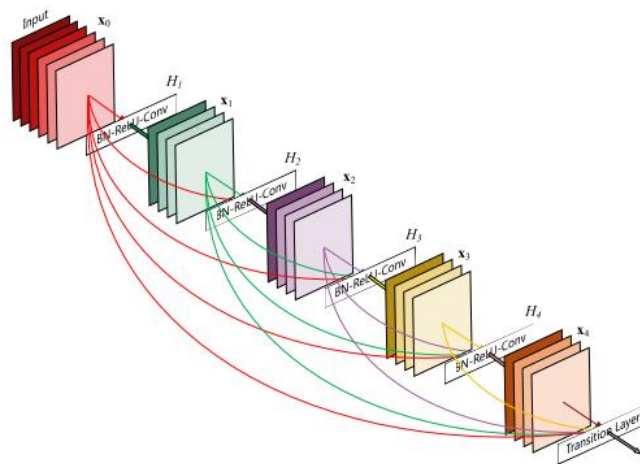


Рисунок 2.8 – приклад архітектури нейромережі DenseNet

Нейромережа DenseNet ефективно бореться з перенавчанням, зменшуючи кількість параметрів і забезпечуючи повторне використання ознак, що підвищує здатність моделі до узагальнення нових даних. Завдяки повторному використанню інформації, мережа може досягти високої продуктивності з меншою кількістю параметрів, ніж інші архітектури. Це дозволяє DenseNet бути економічно ефективною та швидкою при використанні обмежених обчислювальних ресурсів.

#### 2.4.5 Vision Transformers

ViT є відносно новою архітектурою (рис. 2.9) для задач комп'ютерного зору, яка використовує механізми трансформерів замість згорток. ViT розбиває зображення на патчі, перетворюючи їх на послідовність, схожу на обробку тексту в моделях обробки природної мови. Хоча ViT вимагає великих наборів даних для тренування, вона демонструє високу продуктивність, перевершуючи традиційні CNN на деяких задачах.

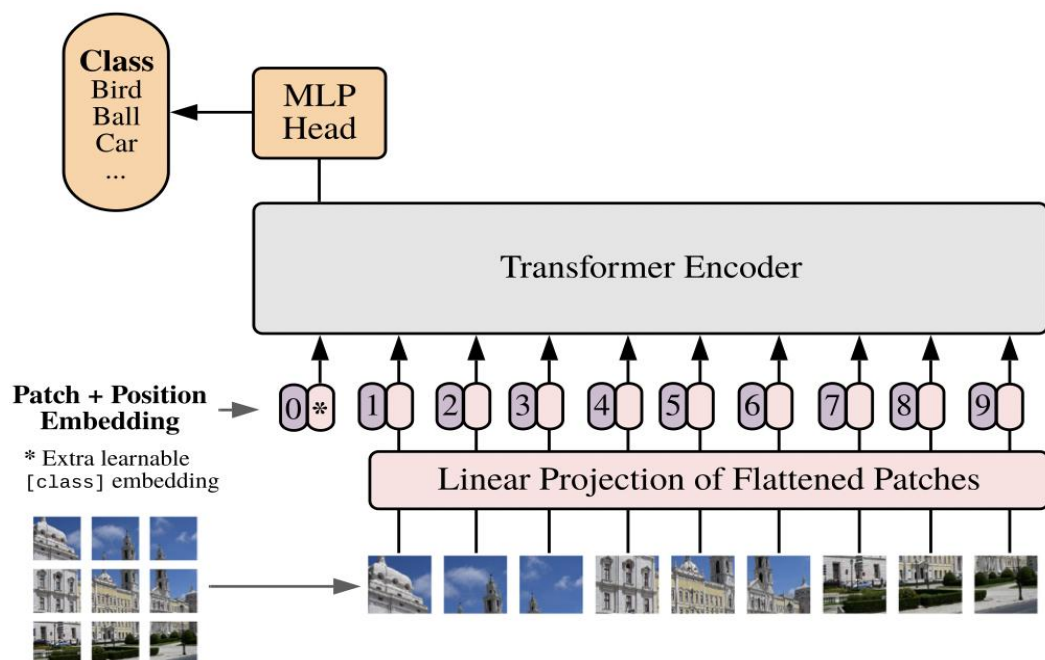


Рисунок 2.9 – архітектура Vision Transformers

Механізми трансформерів (рис. 2.10) замість роботи з окремими пікселями, ViT розбиває зображення на невеликі патчі. Кожен патч обробляється як окремий елемент, подібно до tokenів у тексті, після чого перетворюється у вектор ознак. Це дозволяє моделі працювати з послідовністю патчів, що є більш зручним для трансформерів.

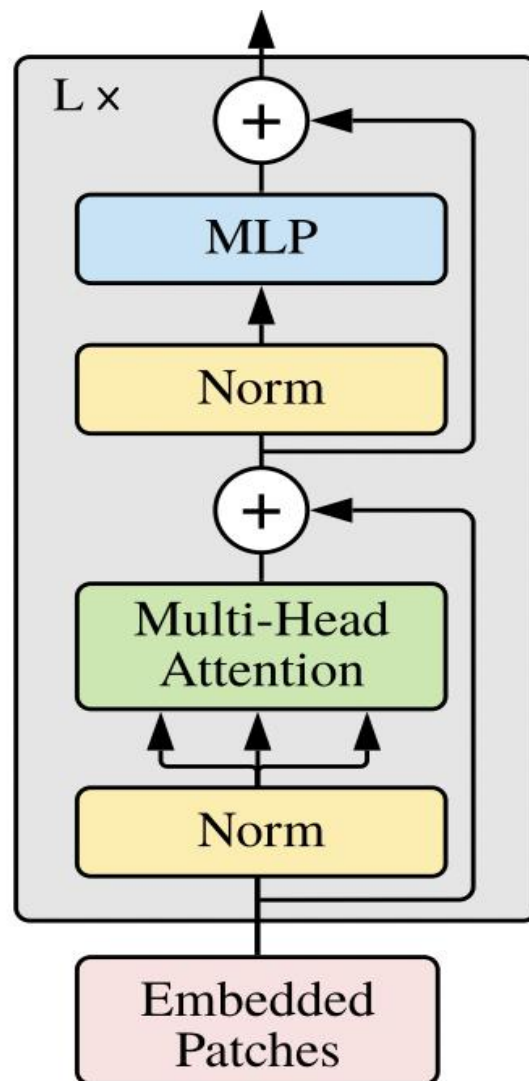


Рисунок 2.10 – схема механізми трансформера

Нейромережа здатна ефективно масштабуватися з ростом кількості шарів і параметрів, що дозволяє створювати надглибокі моделі для складних задач, в тому числі і для класифікації зображень. ViT не потребує попередньо визначеної структури згорток, що робить його гнучкішим при використанні різноманітних типів вхідних даних.

## 2.5 Метрики вимірювання точності моделі

Точність – один із основних показників якості роботи моделі, проте в різних задачах можуть використовуватись й інші метрики. Оцінка точності – це процес вимірювання, наскільки добре модель виконує свою задачу на наборі тестових даних.

Для початку необхідно розподілити дані на тренувальну та тестову вибірки. Тренувальна вибірка використовується для навчання моделі, допомагаючи їй знайти оптимальні ваги і параметри. Тестова вибірка, в свою чергу, це набір даних, який модель ніколи не бачила під час навчання. Використовується для оцінки здатності моделі узагальнювати результати на нові дані.

Як було зазначено раніше, точність – це найпростіша і найпоширеніша метрика для задач класифікації. Вона визначає відсоток правильно передбачених класів з усіх передбачень.

Простим прикладом для пояснення розрахунку точності моделі є класифікація даних на два класи: «Positive» та «Negative». Для оцінки її результатів використовується матриця плутанини (табл. 2.2), яка відображає чотири можливі варіанти передбачень: True Positive (TP) – випадки правильної класифікації позитивного класу; True Negative (TN) – випадки правильної класифікації негативного класу; False Positive (FP) – негативний клас помилково класифікується як позитивний; False Negative (FN) – коли позитивний клас класифікується як негативний.

Таблиця 2.2 – Загальний вид матриці плутанини

|                           | <b>Прогноз: Positive</b> | <b>Прогноз: Negative</b> |
|---------------------------|--------------------------|--------------------------|
| <b>Фактичне: Positive</b> | True Positive (TP)       | False Negative (FN)      |
| <b>Фактичне: Negative</b> | False Positive (FP)      | True Negative (TN)       |

Точність обчислюється за наступною формулою:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2.1)$$

де *Accuracy* – точність.

Припустимо, що модель класифікує 100 зразків. Із них: 50 прикладів – позитивні зразки і ще 50 прикладів це негативні зразки.

Модель правильно класифікувала 40 позитивних як позитивні (TP) і 45 негативних як негативні (TN). Модель помилилася у 5 негативних випадках коли зразки класифіковано як позитивні (FP) і у 10 позитивних класифікованих як негативні (FN). Точність в даному випадку буде розраховуватися наступним чином:

$$Accuracy = \frac{40 + 45}{40 + 45 + 5 + 10} = \frac{85}{100} = 0,85. \quad (2.2)$$

Точність позитивних передбачень знаходять як частку правильних позитивних передбачень серед усіх передбачених позитивних випадків

$$Precision = \frac{TP}{TP + FP}, \quad (2.3)$$

де *Precision* – точність позитивних передбачень.

Чутливість знаходять як частку правильно передбачених позитивних випадків серед усіх справжніх позитивних випадків.

$$Recall = \frac{TP}{TP + FN}, \quad (2.4)$$

де *Recall* – чутливість.

F1-Score використовується в задачах класифікації, коли є необхідність збалансувати точність позитивних передбачень і чутливість, особливо у випадках, коли дані мають дисбаланс класів або коли помилкові позитивні та негативні передбачення мають різні наслідки. F1-Score є гармонійним середнім між точністю та повнотою і показує загальну продуктивність моделі і виглядає наступним чином:

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \quad (2.5)$$

де  $F_1$  – гармонічне середнє між Precision і Recall.

F1-Score також може використовуватись для оцінки моделі, коли потрібно запобігти перенавчанню. Якщо модель має високу точність на тренувальних даних, але низьку повноту на тестових даних, це свідчить про перенавчання. У таких випадках F1-Score допомагає оцінити загальну ефективність моделі.

## 3 ДОСЛІДЖЕННЯ КОМП'ЮТЕРНОЇ МОДЕЛІ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ

### 3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи був розроблений алгоритм для фільтрації зображень за допомогою високочастотних фільтрів. Для реалізації було обране середовище Borland Builder 6.0. Це обумовлено тим, що Builder засновано високопродуктивною мовою C++ [11, 12, 33–41].

Для реалізації було обрано середовище PyCharm 2024.2.1, оскільки воно є одним із найпопулярніших і потужних IDE для програмування на мові Python. PyCharm забезпечує зручну і продуктивну роботу з Python завдяки своїм численним інструментам, таким як автодоповнення коду, інтегроване відлагодження, підтримка віртуальних середовищ і керування пакетами. Інтерфейс застосунку PyCharm показано на рисунку 3.1.

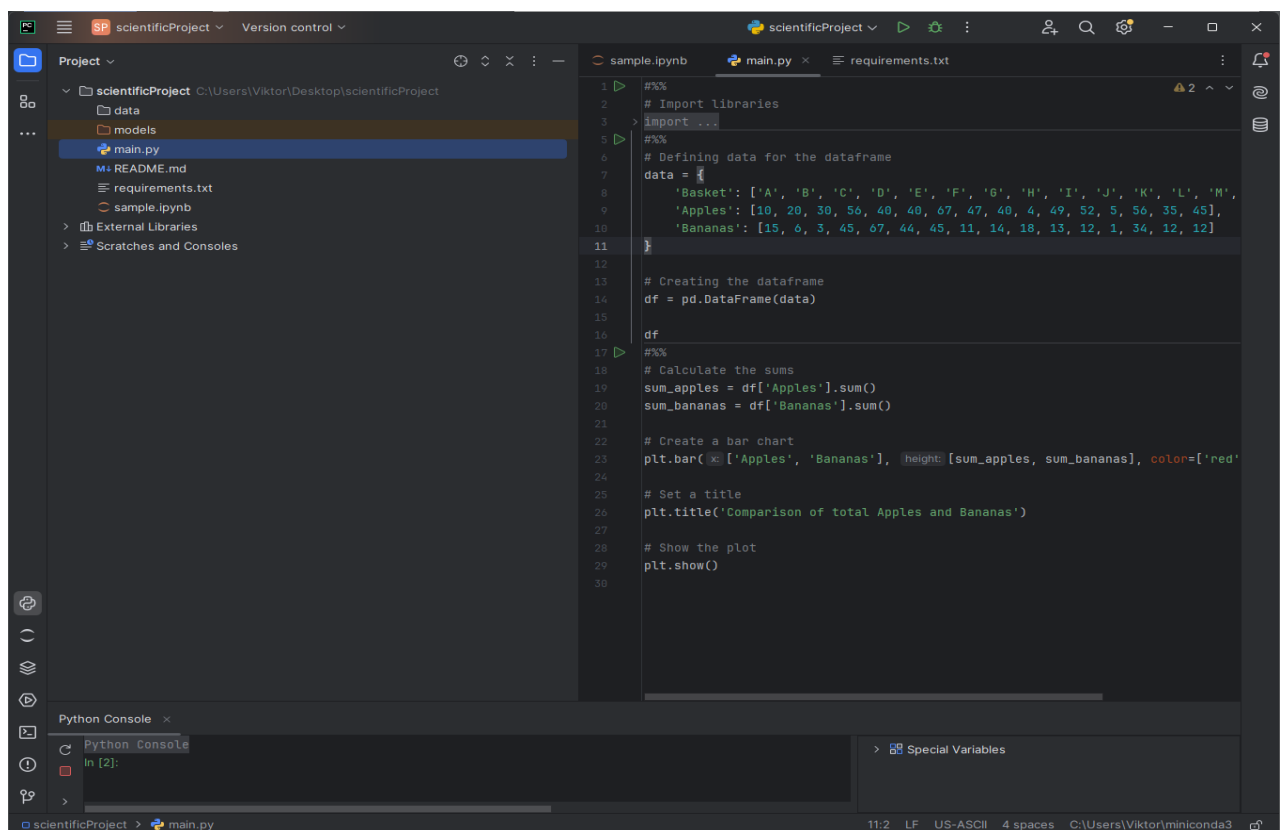


Рисунок 3.1 – Приклад інтерфейсу PyCharm

Це середовище розробки також пропонує функції для роботи з базами даних, фреймворками та науковими бібліотеками, що робить його ідеальним вибором для створення різноманітних застосунків на Python, включаючи десктопні, вебзастосунки та скрипти для автоматизації процесів. Також у середовище розробки PyCharm може бути інтегрований Jupyter Notebook, що дозволяє використовувати всі можливості Jupyter, такі як інтерактивність та візуалізація, у знайомому для розробників середовищі.

Як було зазначено раніше, для створення застосунку було обрано мову Python. Це можна обґрунтувати тим, що Python є потужною і сучасною мовою програмування, яка добре підходить для розробки застосунків, зокрема для класифікації зображень. Вона має простий і зрозумілий синтаксис, що полегшує розробку, а також підтримку об'єктно-орієнтованого програмування. Використання Python дозволить зручно та ефективно реалізувати функціонал класифікації зображень у застосунку, забезпечуючи високу точність і продуктивність моделей.

Для роботи з класифікацією зображень на Python існує велика кількість потужних бібліотек, що робить його ідеальним вибором для реалізації алгоритмів класифікації зображень. Наприклад, бібліотека TensorFlow надає широкий спектр функціональності для побудови, навчання та оцінки нейронних мереж, що використовуються для класифікації зображень. Інші бібліотеки, такі як Keras і PyTorch, також пропонують інструменти для глибокого навчання та аналізу зображень, включаючи підтримку обробки великих наборів даних, налаштування різних моделей, зокрема, ResNet чи EfficientNet, та роботу з попередньо навченими моделями. Використання середовища PyCharm дозволить легко інтегрувати ці бібліотеки в проєкт і ефективно використовувати їх функціонал.

Операційна система Windows надає розробникам додаткові інструменти для роботи з Python, зокрема через бібліотеки, які підтримують апаратне прискорення, наприклад, CUDA для NVIDIA GPU, що є критично важливим для виконання обчислювальних задач, зокрема в галузі машинного навчання,



обробки зображень та інших ресурсномістких обчислень. У мові розробки Python доступні прості та ефективні засоби для роботи з класифікацією зображень, які дозволяють швидко створювати, навчати та використовувати моделі машинного навчання.

### 3.1.1 Jupyter Notebook

Jupyter Notebook – це інтерактивне середовище для аналізу даних та виконання програмного коду, яке особливо популярне серед науковців, аналітиків даних та розробників машинного навчання. Воно дозволяє виконувати код поетапно та отримувати миттєвий зворотний зв'язок, що робить його ідеальним для експериментів з даними, візуалізацій і моделювання. Взаємодія між Python і Jupyter Notebook побудована на основі так званих «ядер», що виконують код Python і повертають результати безпосередньо у середовище.

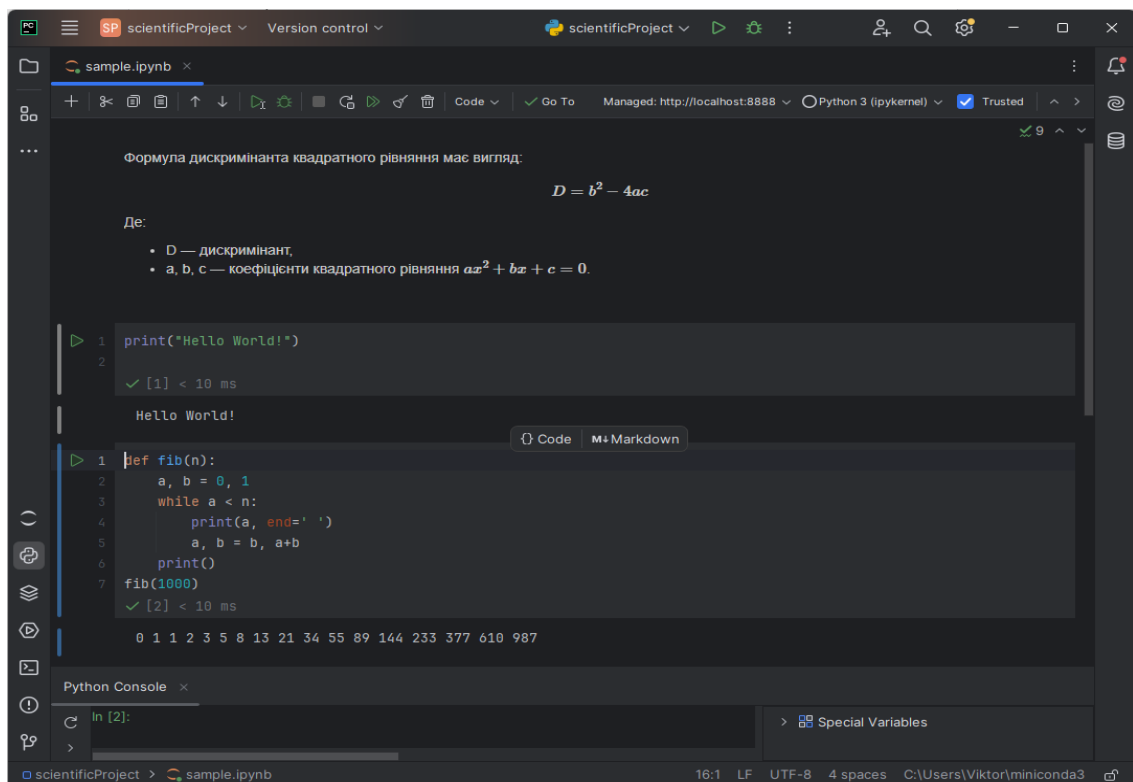
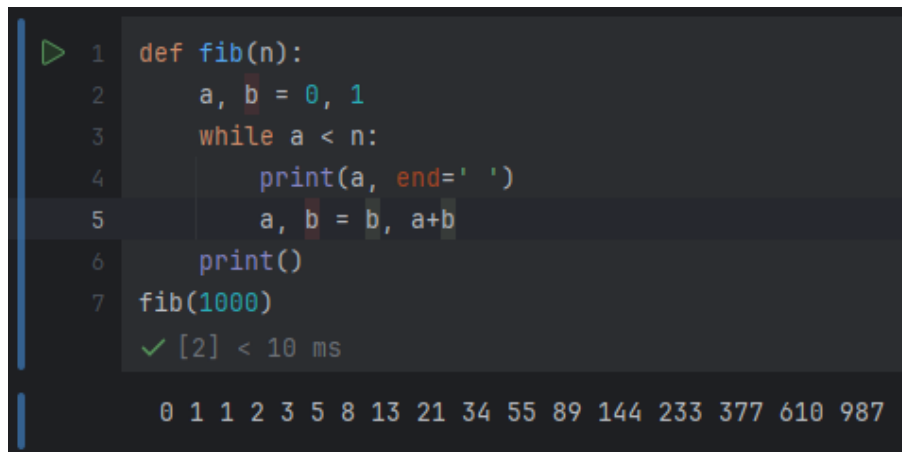


Рисунок 3.2 – інтерфейс додатку у Jupyter Notebook

Jupyter Notebook має простий і інтуїтивно зрозумілий інтерфейс, який складається з кількох ключових компонентів, але основний вміст Jupyter Notebook організовано в комірках. Комірки поділяють на два види: комірки для коду та комірки для розмітки.

В комірках для коду можна писати та виконувати код на Python. Коли запускається комірка, результат її виконання з'являється безпосередньо під коміркою. Оскільки можна запускати окремі комірки коду, а не весь код одразу, то це дає змогу легко експериментувати з кодом і вносити зміни в реальному часі. Комірку для написання коду показано на рисунку 3.3.



```

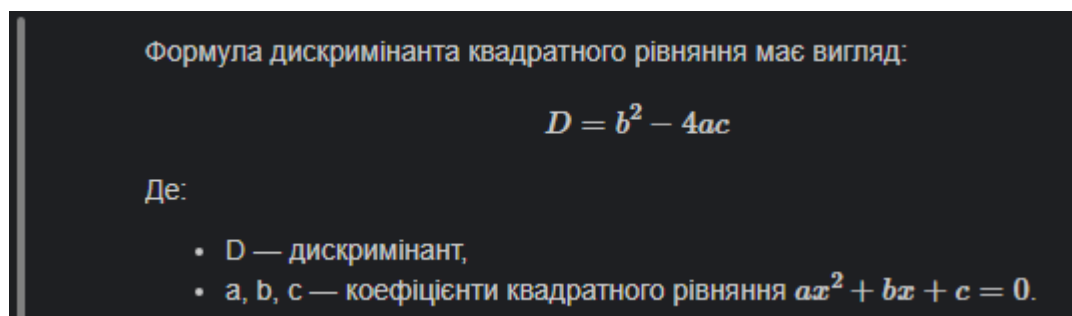
1 def fib(n):
2     a, b = 0, 1
3     while a < n:
4         print(a, end=' ')
5         a, b = b, a+b
6     print()
7 fib(1000)
✓ [2] < 10 ms

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987

```

Рисунок 3.3 – комірка для написання коду

В комірках для розмітки зазвичай описують документацію коду. У таких комірках можна використовувати форматування, заголовки, списки, зображення та навіть LaTeX для математичних рівнянь. Комірку для розмітки показано на рисунку 3.4.



Формула дискримінанта квадратного рівняння має вигляд:

$$D = b^2 - 4ac$$

Де:

- $D$  — дискримінант,
- $a, b, c$  — коефіцієнти квадратного рівняння  $ax^2 + bx + c = 0$ .

Рисунок 3.4 – комірка для розмітки

### 3.1.2 Бібліотеки для реалізації алгоритмів машинного навчання

## 3.2 Програмна реалізація

Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» (рис. А.1).

## 3.3 Інструкція користувача

Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» (рис. А.2).

## 3.4 Тестування розробленої моделі

Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» (рис. А.3).

## ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований метод покращення зображень за допомогою

Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст».

Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст».

Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст» Текст стилем «Основний текст».

Результати дослідження апробовано у вигляді 3 тез доповідей під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [42], XV Регіональної студентської науково-технічної конференції «НАУКА – ПЕРШІ КРОКИ» [43], XXVI Міжнародної науково-практичної конференції «Topical issues of practice and science» [44].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

### Приклад, як оформити навчальний посібник або монографію:

1. Путятін, Є. П., Гороховатський, В. О., & Матат, О. О. (2006). Методи та алгоритми комп'ютерного зору: навч. посібник.
2. Творошенко, І. С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. *Харків: ХНУРЕ*.
3. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.
4. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень: навч. посібник. *Харків: ХНУРЕ*.
5. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.
6. Best Practices for Image Preprocessing in Image Classification. URL: <https://keylabs.ai/blog/best-practices-for-image-preprocessing-in-image-classification/> (дата звернення 22.10.2024).

### Приклад, як оформити статтю у журналі:

7. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.
8. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.
9. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to

training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

10. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.

11. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.

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

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

14. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069-6084.

15. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.

16. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.

17. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.

18. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

19. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

20. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

21. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.

22. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.

23. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

24. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

25. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.

**Приклад, як оформити тези конференції:**

26. Gorokhovatskyi, V., Gorokhovatskyi, O., Yevgenyi, P., & Olena, P. (2018). Quantization of the Space of Structural Image Features as a Way to.

27. Gorokhovatskyi, V., & Tvoroshenko, I. (2020). Image Classification Based on the Kohonen Network and the Data Space Modification.

28. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.

**Приклад, як оформити електронне посилання:**

29. The flow of improved BRISK algorithm. URL: [https://www.researchgate.net/figure/the-flow-of-improved-BRISK-algorithm\\_fig1\\_328946366](https://www.researchgate.net/figure/the-flow-of-improved-BRISK-algorithm_fig1_328946366) (дата звернення 14.10.2023).

**Приклад, як оформити посилання на свою попередню кваліфікаційну роботу:**

30. Іванов І.І. Дослідження методів пошуку зображення у базах даних: кваліфікаційна робота першого (бакалаврського) рівня вищої освіти: 122 Комп'ютерні науки. Харків, 2023. 75 с.

**Увага!!! Усі посилання на джерела повинні бути у форматі APA!!!**

**Щоб правильно оформити посилання, рекомендуємо скористатися <https://scholar.google.com.ua>. Можна знайти необхідну літературу за автором чи тематикою, вибрати процитовану літературу, а потім за допомогою перейти до трьох видів форматів та скопіювати текст формату APA.**



У цьому розділі має бути не менше 40 джерел інформації,  $\frac{3}{4}$  із них – за останні 5 років, обов'язкова наявність публікацій керівника кваліфікаційної роботи.

Забороняється цитування в тексті та внесення до бібліографічних списків тих джерел, які опубліковані російською мовою в будь-якій країні, а також джерел іншими мовами, якщо вони опубліковані на території росії та білорусі. Тобто в перелік джерел посилання НЕ вносимо джерела на російській мові (навіть ваших керівників кваліфікаційних робіт!!!), а також, якщо видавництво публікації розташоване на вказаних вище територіях.

## ДОДАТОК А

## ТЕСТОВІ ЗОБРАЖЕННЯ

A photograph of a handwritten word 'people,' in cursive script on a light-colored, slightly textured paper. The word is written in dark ink and is slightly slanted to the right.

Рисунок А.1 – Приклад рукописного тексту № 1

A photograph of a handwritten word 'meeting' in cursive script on a light-colored, slightly textured paper. The word is written in dark ink and is slightly slanted to the right.

Рисунок А.2 – Приклад рукописного тексту № 2

A photograph of a handwritten phrase 'as a percentage of social service' in cursive script on a light-colored, slightly textured paper. The phrase is written in dark ink and is slightly slanted to the right.

Рисунок А.3 – Приклад рукописного тексту № 3