

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет імені Петра Могили**

**Факультет комп'ютерних наук**

**Кафедра інженерії програмного забезпечення**

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_ Є. О. Давиденко  
*підпис*

«\_\_» \_\_\_\_\_ 2024р.

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**Програмне забезпечення розпізнавання та сегментації  
об'єктів на зображеннях з використанням  
комп'ютерного зору**

Спеціальність «Інженерія програмного забезпечення»

121 – КРМ.1 – 608м.21810814

*Здобувач вищої освіти*

\_\_\_\_\_ М. Д. Забеленков  
*підпис*

«\_\_» \_\_\_\_\_ 2024р.

**Керівник PhD, ст. викладач**

\_\_\_\_\_ І. О. Кандиба  
*підпис*

«\_\_» \_\_\_\_\_ 2024р.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет ім. П. Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Є. О. Давиденко

«\_\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи магістра**

Видано студенту групи б08м факультету комп'ютерних наук

Забеленкову Максиму Дмитровичу

*(прізвище, ім'я, по батькові студента)*

1. Тема кваліфікаційної роботи

«Програмне забезпечення розпізнавання та сегментації об'єктів на зображеннях з використанням комп'ютерного зору»

Затверджена наказом по ЧНУ від «10» листопада 2023 р. № 234

2. Строк представлення кваліфікаційної роботи «\_\_\_\_» \_\_\_\_\_ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні:

Очікуваним результатом роботи є програмне забезпечення розпізнавання та сегментації об'єктів на зображеннях з використанням комп'ютерного зору

4. Перелік питань, що підлягають розробці: огляд технологій комп'ютерного зору; визначення основних понять розпізнавання та сегментації; огляд методів розпізнавання об'єктів; огляд методів сегментації; обґрунтування вибору методів дослідження; специфікація вимог до програмного застосунку; моделювання та проектування програмного забезпечення застосунку; вибір інструментів для реалізації програмного забезпечення; програмна реалізація та тестування програмного забезпечення; написання документації.

5. Перелік графічних матеріалів

Слайди презентації

---

Керівник роботи PhD,ст. викладач Кандиба І. О.

---

*(посада, прізвище, ім'я, по батькові)*

\_\_\_\_\_  
*(підпис)*

Завдання прийнято до виконання

Забеленков Максим Дмитрович

---

*(прізвище, ім'я, по батькові студента)*

Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 2023 р

\_\_\_\_\_  
*(підпис)*

## КАЛЕНДАРНИЙ ПЛАН Виконання кваліфікаційної роботи

**Тема:** «Програмне забезпечення розпізнавання та сегментації об'єктів на зображеннях з використанням комп'ютерного зору»

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання	12.11.2023	19.11.2023	Виконано
2.	Огляд літератури за темою роботи	20.11.2023	01.12.2023	Виконано
3.	Складання календарного плану КРМ	02.12.2023	03.12.2023	Виконано
4.	Огляд технологій комп'ютерного зору	04.12.2023	10.12.2023	Виконано
5.	Огляд методів розпізнавання та сегментації	11.12.2023	18.12.2023	Виконано
6.	Аналіз ринку наявних технологій і рішень для вирішення завдань	19.12.2023	26.12.2023	Виконано
7.	Розробка вимог програмного забезпечення	27.12.2023	30.12.2023	Виконано
8.	Проектування архітектури програмного забезпечення	05.01.2024	10.01.2024	Виконано
9.	Реалізація та тестування програмного забезпечення	11.01.2024	25.01.2024	Виконано
10.	Відгук керівника КРМ	03.02.2024	03.02.2024	Виконано
11.	Оформлення КРМ та презентації	04.02.2024	07.02.2024	Виконано
12.	Попередній захист	08.02.2024	08.02.2024	Виконано
13.	Рецензування	11.02.2024	12.02.2024	Виконано
14.	Завершення оформлення КРМ та презентації	13.02.2024	15.02.2024	Виконано
15.	Захист кваліфікаційної роботи магістра	26.02.2024	26.02.2024	Виконано

Розробив студент Забелєнков Максим Дмитрович

*(прізвище, ім'я, по батькові студента)*

*(підпис)*

«        » \_\_\_\_\_ 2023 р.

Керівник роботи PhD, ст. викладач ПЗ Кандиба Ігор Олександрович

*(посада, прізвище, ім'я, по батькові)*

*(підпис)*

«        » \_\_\_\_\_ 2023 р.

## АНОТАЦІЯ

до кваліфікаційної роботи магістра

«Програмне забезпечення розпізнавання та сегментації об'єктів на зображеннях з використанням комп'ютерного зору»

Студент 608м гр. : Забеленков Максим Дмитрович

Керівник: PhD, ст. викладач Кандиба І. О

Розпізнавання та сегментація об'єктів на зображеннях є ключовими завданнями у сфері обробки зображень та комп'ютерного зору.

**Об'єкт роботи:** процес розпізнавання та сегментації об'єктів на зображеннях з використання комп'ютерного зору.

**Предмет роботи:** інструментальні засоби розробки програмного забезпечення розпізнавання та сегментації об'єктів.

**Мета і завдання роботи:** вдосконалити процес розпізнавання та сегментації об'єктів на зображеннях шляхом створення програмного забезпечення з використанням інструментарію штучного інтелекту.

Для досягнення мети було виконано перелік наступних завдань:

1. оглянуто технологій комп'ютерного зору;
2. досліджено методи розпізнавання та сегментації;
3. проаналізовано ринок наявних технологій і рішень для вирішення завдань розпізнавання та сегментації зображень;
4. спроектовано архітектуру програмного забезпечення;
5. досліджено специфікації вимог;
6. обрано інструменти для реалізації застосунку;
7. програмно реалізовано застосунок;
8. протестовано програму розпізнавання.

У першому розділі розглянуто предметну сферу та головні можливості методів розпізнавання та сегментації, досліджено проблематики, а також наведено функціональні вимоги. У другому розділі розглянуто математичну модель сегментації та її методів, також визначено архітектури згорткових нейронних мереж та порівняно їх для визначення найкращої. У третьому розділі описані

програмні методи реалізації застосунку, обрані датасети, а також навчання моделей, відповідними даними. У четвертому розділі розроблено та протестовано програмний код застосунку.

У результаті було розроблено застосунок для розпізнавання та сегментації зображень.

Сторінок – 89, таблиць – 2 , рисунків – 48 , посилань – 24 , додатків – 2 .

*Ключові слова: сегментація, нейромережа, комп'ютерний зір, CNN, YOLO, датасет.*

## ABSTRACT

to the master's qualification work

"Software for Object Recognition and Image Segmentation  
Using Computer Vision"

Student of group 608m: Zabielenkov Maksym

Supervisor: PhD, Senior Lecturer Kandyba I. O.

Recognition and segmentation of objects in images are key tasks in the field of image processing and computer vision.

**Object of the work:** the process of recognizing and segmenting objects in images using computer vision.

**The subject of the work:** tools for development of object recognition and segmentation software.

**The purpose and tasks of the work:** to improve the process of recognition and segmentation of objects in images by creating software using artificial intelligence tools.

To achieve the goal, a list of the following tasks was defined:

1. reviewed computer vision technologies;
2. methods of recognition and segmentation were investigated;
3. the market of available technologies and solutions for solving image recognition and segmentation tasks was analyzed;
4. the software architecture is designed;
5. specifications of requirements were studied;
6. tools for application implementation are selected;
7. the application is implemented programmatically;
8. the recognition program was tested.

In the first chapter, the subject area and main capabilities of the recognition and segmentation methods are considered, the problems are investigated, and the functional requirements are given. The second chapter examines the mathematical model of segmentation and its methods, also defines the architecture of convolutional neural networks and compares them to determine the best one. The third chapter describes software implementation methods of the application, selected datasets, as well as training of models with relevant data. In the fourth chapter, the application

code is developed and tested.

As a result, an application for image recognition and segmentation was developed.  
Pages – 89 , tables – 2 , figures – 48 , links – 24, appendices – 2.

*Keywords: segmentation, neural network, computer vision, CNN, YOLO, dataset.*



## Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ЗАСОБІВ РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ З ВИКОРИСТАННЯМ КОМП'ЮТЕРНОГО ЗОРУ .....	8
1.1 Опис процесу розпізнавання та сегментації зображень за допомогою комп'ютерного зору .....	8
1.2 Проблематика, методи обробки та розпізнавання зображень .....	11
1.3 Огляд існуючих рішень .....	16
1.4 Специфікація вимог програмного забезпечення .....	22
Висновки до розділу 1.....	23
2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ .....	24
2.1 Математична модель сегментації зображень.....	25
2.3 Оператори згладжування зображення .....	30
2.3 Архітектури згорткових нейронних мереж.....	31
2.4 Порівняння методів згорткових нейронних мереж .....	35
Висновок до розділу 2.....	39
3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ .....	40
3.1 Програмні засоби для розпізнавання, сегментації зображень .....	40
3.2 Архітектура програмного забезпечення розпізнавання та сегментації об'єктів.....	44
3.3 Аналіз та огляд наборів даних для розпізнавання та сегментації зображень	46
3.3 Визначення алгоритму навчання моделі YOLO .....	51
3.4 Метрики сегментації .....	53
Висновок до розділу 3.....	56
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ З СЕГМЕНТАЦІЇ ТА РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ.....	57
4.1 Розробка структури застосунку .....	57
4.2 Кодування програмного застосунку .....	58

4.3 Тестування програмного забезпечення.....	63
Висновки до розділу 4.....	69
ВИСНОВКИ .....	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	71
ДОДАТОК А Матеріали апробації роботи .....	73
ДОДАТОК Б Лістинг програми .....	78

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

CNN – Згортова нейронна мережа (Convolutional neural network)

YOLO – You Look Only Once

NumPy – Numerical Python

HOG - Histogram of Oriented Gradients

RNN – Рекурента нейронна мережа (Recurrent neural network)

OpenCV - OpenSource Computer Vision

ПЗ – програмне забезпечення

## ВСТУП

У сучасному світі розвиток автоматизації та роботизації процесів торкнувся усіх сфер життя людини. Промисловість, наука, побут, сільське господарство та тощо, вже доволі великий проміжок часу приділяють увагу розробці чи використанню програмного забезпечення різного роду. Його використання надає можливість значно оптимізувати та підвищити продуктивність тих чи інших результатів в залежності від необхідного результату.

Наразі ПЗ використовується у більшості галузях світу та вже давно стали частиною нашого побуту. Медицина та лінгвістика, криміналістика та ядерна наука, військова справа та системи управління, тощо. Це лише невелика частка від усієї сукупності галузей. З кожним днем кількість залучених галузей росте завдяки поточному технічному прогресу.

Розвиток штучного інтелекту не є випадковістю, а є цілком логічним витком з розвитку програмного забезпечення. Наразі застосування штучного інтелекту набирає неабияких обертів, створення віртуальних асистентів, систем автоматичного керування та тощо. Але так чи інакше штучний інтелект спирається вже на наявну інформацію, саме тому дуже важливо вміти її обробляти та використовувати.

Зображення – це один з методів передачі інформації, який з'явився багато-багато століть назад, але з часом та розвитком технологій з'явилася можливість здобуття інформації з зображень, що стало одним з головних методів її обробки. Саме, тому розробка програмного забезпечення для розпізнавання та сегментації об'єктів є доволі важливою та актуальною.

Розпізнавання та сегментація об'єктів на зображеннях стали ключовими завданнями у сфері обробки зображень та комп'ютерного зору.

Комп'ютерний зір (Computer Vision) — це галузь науки та технологій, яка вивчає та розробляє системи, призначені для надання комп'ютерам здатності інтерпретувати та розуміти візуальні дані зі світу, подібно тому, як це робить людський зір.

Основні завдання комп'ютерного зору включають в себе:

1. розпізнавання об'єктів: автоматичне визначення та класифікація об'єктів на зображеннях чи в відеопотоках;
2. сегментація зображень: виділення та класифікація окремих областей на зображеннях, що дозволяє розрізняти різні об'єкти та їхні межі;
3. визначення руху: виявлення та відстеження рухомих об'єктів в часі;
4. розпізнавання обличчя: Виявлення та ідентифікація обличчя на зображеннях чи відеопотоках;
5. доступ до глибини: вимірювання відстані до об'єктів на зображенні для отримання тривимірної інформації.

Для досягнення цих завдань використовуються різні методи, включаючи класичні методи обробки зображень та сучасні підходи, такі як глибоке навчання (зокрема, згорткові нейронні мережі

Цей напрямок розвитку технологій має ряд вагомих переваг і застосувань. Однією з них є відомості, які можна отримати з ретельного аналізу зображень. Програмне забезпечення для розпізнавання об'єктів дозволяє ідентифікувати та класифікувати об'єкти на зображеннях, що є важливим в багатьох галузях, таких як медицина, виробництво та автономні транспортні засоби.

Сегментація зображень в свою чергу дозволяє виділити та визначити конкретні області або об'єкти на зображенні, що важливо для подальшого аналізу та використання цільової інформації. Це знаходить застосування в медичних дослідженнях, виробництві та в інших галузях, де важлива точність та деталізація результатів.

Розробка програмного забезпечення для цих завдань вимагає використання сучасних методів глибокого навчання, таких як згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN). Ці технології дозволяють автоматизувати процес розпізнавання та сегментації, а також покращують точність результатів.

Рекурентні нейронні мережі (RNN) та згорткові нейронні мережі (CNN) є двома популярними архітектурами глибокого навчання, які використовуються для обробки послідовностей та аналізу зображень відповідно.

В сучасному світі важливо також враховувати питання етики та безпеки у

використанні цих технологій, забезпечуючи конфіденційність та захист особистої інформації. Таким чином, розробка програмного забезпечення для розпізнавання та сегментації об'єктів на зображеннях є важливим етапом у високотехнологічному суспільстві, сприяючи автоматизації та удосконаленню процесів обробки візуальної інформації.

**Об'єкт роботи:** процес розпізнавання та сегментації об'єктів на зображеннях з використання комп'ютерного зору.

**Предмет роботи:** інструментальні засоби розробки програмного забезпечення розпізнавання та сегментації об'єктів.

**Мета і завдання роботи:** вдосконалити процес розпізнавання та сегментації об'єктів на зображеннях шляхом створення програмного забезпечення з використанням інструментарію штучного інтелекту.

Для досягнення мети було виконано перелік наступних завдань:

9. оглянуто технологій комп'ютерного зору;
10. досліджено методи розпізнавання та сегментації;
11. проаналізовано ринок наявних технологій і рішень для вирішення завдань розпізнавання та сегментації зображень;
12. спроектовано архітектуру програмного забезпечення;
13. досліджено специфікації вимог;
14. обрано інструменти для реалізації застосунку;
15. програмно реалізовано застосунок;
16. протестовано програму розпізнавання.

# **1 АНАЛІЗ ЗАСОБІВ РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ З ВИКОРИСТАННЯМ КОМП'ЮТЕРНОГО ЗОРУ**

## **1.1 Опис процесу розпізнавання та сегментації зображень за допомогою комп'ютерного зору**

Комп'ютерний зір – це галузь, яка об'єднує методи обробки, аналізу та розпізнавання зображень, а також великі обсяги даних з фізичного виміру, з метою отримання інтерпретованих результатів у чисельній або символній формі.

Основне призначення комп'ютерного зору полягає у можливості моделювання властивостей людського зору, зокрема у здатності розпізнавати образи на зображеннях і витягати з них необхідну інформацію.

Комп'ютерний зір також є науковою дисципліною, що ґрунтується на теорії створення та використання моделей та систем для виділення інформації з зображень. Ця галузь не обмежується лише статичними зображеннями, а також включає в себе відеодані, тривимірні дані та інші види вхідної інформації. Комп'ютерний зір знаходить широке застосування в розробці систем для автоматизованого розпізнавання зображень та відеоданих.

Розпізнавання та сегментація завжди є ключовими аспектами взаємодії комп'ютерного зору, спрямованими на аналіз та обробку статичних та динамічних зображень. Ці технології вже широко використовуються в різних сферах, таких як аналіз медичних зображень, автоматичний моніторинг безпеки та інші. Таким чином, багато фахівців у сучасній інформаційній сфері проводять глибокі дослідження в галузі сегментації зображень та алгоритмів багатоцільового розпізнавання, сподіваючись на подальше удосконалення їх продуктивності в усіх напрямках.

Розпізнавання об'єктів та образів – це розділ теорії штучного інтелекту, що вивчає та досліджує методи класифікації об'єктів. Об'єкти називають “образами”, який може представляти собою літеру або цифру, запис мови або цифрову фотографію, тощо. Ця галузь входить в склад теорії штучного інтелекту, а також є частиною – теорії машинного навчання. Головна мета теорії машинного навчання – це створення алгоритмів, які маю змогу навчатися.

Розрізняють два головних підходи до навчання – це індуктивний та дедуктивний. Базою індуктивного навчання, або як його ще називають, навчання з прецедентами, є виявлення загальних властивостей об'єктів на основі обмеженої кількості інформації, що здобувається за допомогою емпіричних методів. Дедуктивне навчання базується на формалізації сукупності знань експертів у вигляді типових баз знань, наприклад експертні системи.

Розпізнавання образів побудовано на використанні сукупностей математичного апарату, який включає в себе наступні дисципліни: математична статистика, методи оптимізації, дискретну математику, алгебру та геометрію.

Розпізнавання образів широко використовується у розробці всіх комп'ютерних систем, які потребують інтелектуальних функцій, тобто таких, що пов'язані з умінням приймати рішення на заміну людині. Це охоплює такі області, як медична діагностика, криміналістична експертиза, військова справа, пошук інформації та аналіз даних.

Розглядаючи сутність образу, можна зазначити, що він є набором чисел, які описують його властивості та характеристики, що ще називають “ознаками”. Коли “ознаки” упорядковуються у відповідний набір вони називаються – вектором ознак.

Розглядаючи задачу розпізнавання образів, можна виокремити певні підзадачі.

1. Генерування ознак (feature generation) – отримання чи обчислення числових ознак, які описують характеристики об'єктів;
2. Вибір ознак (feature selection) – визначення найбільш інформативних ознак для класифікацій, включаючи їх первинні ознаки та функції;
3. Побудова класифікатора (classifier construction) – створення вирішального та головного правила, яке буде виступати основою для класифікації об'єктів;
4. Оцінка якості класифікації (classifier estimation) – розрахунок показників вірності класифікатора, а саме: точність, чутливість, специфічність та оцінка помилок.



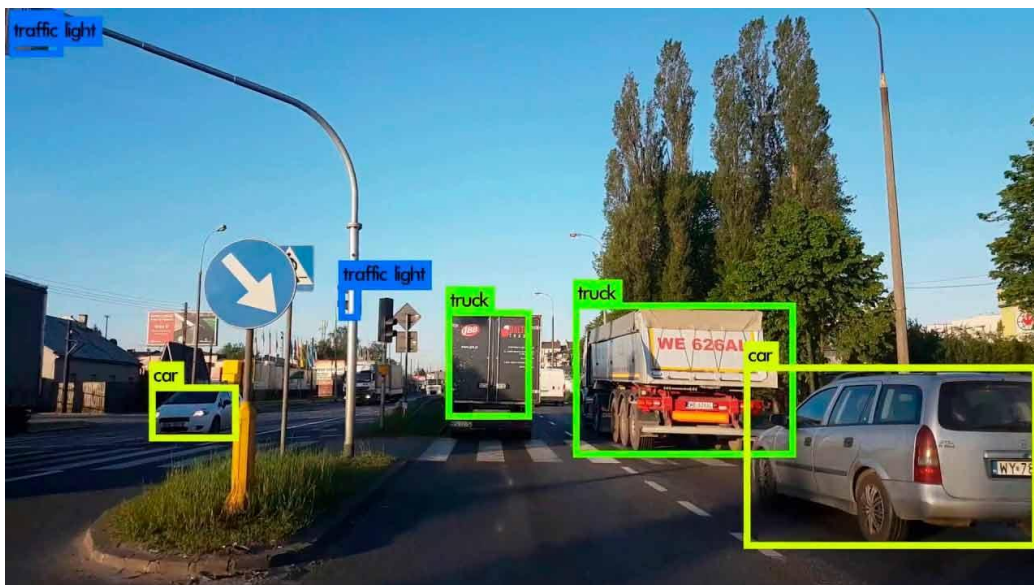


Рисунок 1.1 – Розпізнавання об'єктів на зображенні

Сегментація зображення є однією з основних задач комп'ютерного зору, яка насамперед розширює передумову виявлення об'єктів.

Сегментація – це один з базових методів обробки та аналізу, який використовується для розбиття зображення на певні частини та області, які ще називають сегментами. Це є певним процесом присвоєння кожному пікселю мітки. Пікселі з однаковою міткою мають подібні візуальні характеристики. За рахунок цього виділяються тільки необхідні та найважливіші об'єкти зображення.

Найчастіше основа сегментації лежить на властивостях пікселів зображення. Базова сегментація включає в себе як і відокремлення переднього плану, так й розбиття на кластери зображення, які мають основу співпадіння форм або спектру кольорів.

Кількість алгоритмів та методів сегментації зростає з кожним роком, але можна виділити деякі основи з них, а саме: порогова, на основі країв, на основі регіонів, на основі кластеризації, на основі штучної нейронної мережі та за допомогою алгоритмів сегментації. Усі ці методи наведлені на наступному рисунку.



Рисунок 1.2 – Алгоритми сегментації

## 1.2 Проблематика, методи обробки та розпізнавання зображень

Головним та першочерговим завдання методів та алгоритмів обробки зображень є абсолютно чітке вихідне зображення. Але так чи інакше існують різноманітні умови за яких складність виконання обробки збільшується у рази. До таких умов відносяться:

1. перешкоди та шум;
2. фон зі складною структурою, перекривання чи накладання об'єктів один на одний;
3. складно оброблювальні ракурси;
4. розфокусування;
5. дисторсії об'єктів;
6. великий спектр та різка зміна графічних характеристик;
7. дефекти;
8. зміна середовища (розсіювання світла, туман, засвітлення сонячним промінням, тощо).

Для боротьби з проблемами, що виникають було створено багато різноманітних методів для обробки та розпізнавання зображень.

Методи займають дуже суттєве та важливе місце у сучасній науці та галузі комп'ютерного зору, так чи інакше під обробкою графічних даних розуміють ще й класифікація об'єктів за характеристиками та властивостями.

### **Класичні методи розпізнавання та обробки зображень**

Обробка зображень на комп'ютері стає можливою лише після перетворення неперервного сигналу зображення у цифрову форму. Ефективність цього процесу значно залежить від того, наскільки адекватною є модель, що використовується для опису зображення і необхідна для розробки відповідних алгоритмів обробки. Модель зображення представляє собою систему функцій, які описують ключові характеристики зображення, такі як функція яскравості, яка відображає зміни яскравості в площині зображення, а також просторові та спектральні характеристики.

Кожен канал зображення включає оптичну систему, оптоелектричний перетворювач, пристрій для аналого-цифрового перетворення (АЦП) та обчислювальний блок для цифрової обробки сигналів.

### **Статистичні методи розпізнавання та обробки зображень**

Статистичні методи у свої основі містять статистичні підходи. Ці підходи базуються на обробці та перетворенні пікселів, областей зображення. До головних аспектів статистичного аналізу входять:

1. середні значення та дисперсія, що дозволяє визначити інформацію про яскравість та варіабельність зображення;
2. гістограма, відображення базового розподілу яскравості пікселів зображення;
3. кореляція, дозволяє визначити ступені взаємозв'язку між різними фрагментами зображення;
4. моменти зображення, для більшої обробки форм та структур об'єктів;
5. фільтрація за різноманітними характеристиками;
6. статистичні тестування для перевірки визначених гіпотез у розбіжностях чи співпадіннях певних областей зображення.

## **Методи розпізнавання та обробки зображень на основі динамічних моделей**

Методи розпізнавання та обробки зображень на основі динамічних моделей використовують концепції динаміки та зміни в часі для аналізу та розуміння зображень. Ці методи дозволяють враховувати динаміку об'єктів та їхню поведінку в часі, що є особливо важливим у випадках, коли об'єкти змінюють своє положення, форму чи орієнтацію протягом часу.

Деякі ключові аспекти методів на основі динамічних моделей включають:

1. трекінг об'єктів, що базується на спостереженні та відстеженні рухів об'єктів на зображеннях протягом часу;
2. моделювання руху в основі якого є використання математичних моделей для опису динаміки та руху об'єктів;
3. виявлення змін аналізує зміни в часі для виявлення аномалій або важливих подій на зображеннях;
4. аналіз часових рядів що використовує методи обробки сигналів та статистичних підходів для аналізу та витягування інформації з часових рядів зображень;

Ці методи знаходять застосування у великому спектрі областей, включаючи відеоспостереження, медичне зображення, робототехніку та інші. Застосування динамічних моделей дозволяє отримати більш повне та деталізоване розуміння змін на зображеннях в часі.

### **Фрактальні методи**

Фрактальні методи – це методи, які побудовані на ідеї фракталів і використовують їхні властивості для обробки та розпізнавання зображення. Об'єкти, створені людьми, такі як промислові та житлові будівлі, можуть бути ефективно описані за допомогою простих геометричних форм, таких як куби, сфери, циліндри, конуси. Однак природні текстури, які є нерегулярними та фрагментарними, важко піддаються такому опису. У зв'язку з цим природним є представлення їх у вигляді фракталу з певною розмірністю  $D$ .

Фрактал - це самоподібна множина нецілої розмірності, що складається з об'єднання однакових непересічних підмножин, схожих на вихідну множину.

Основні властивості фракталів включають тонку структуру на довільно малих масштабах, нерегулярність, форму самоподібності та фрактальну розмірність Мінковського.

Фрактали, як правило, класифікуються за їхнім походженням та математичною природою. Одна з основних класифікацій включає геометричні, алгебраїчні та стохастичні фрактали.

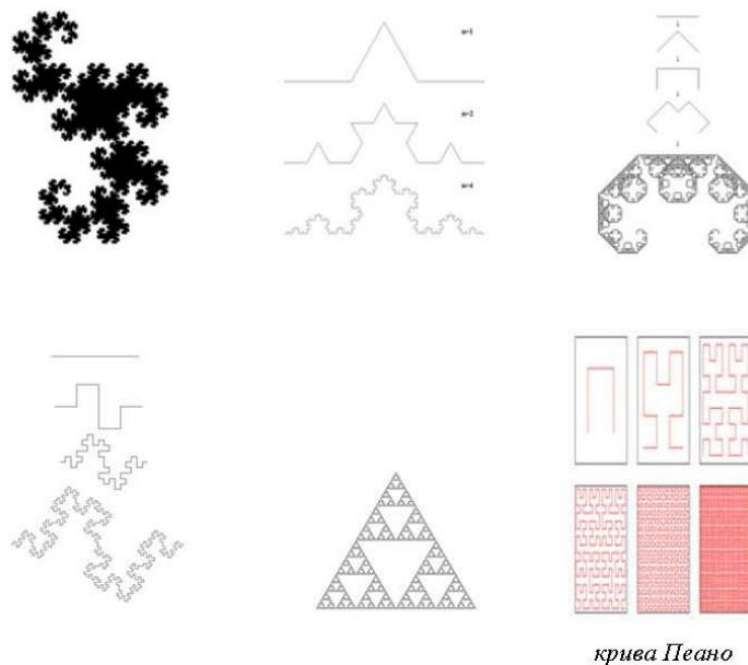


Рисунок 1.3 – Геометричні фрактали

Геометричні фрактали визначаються за допомогою геометричних перетворень та самоподібності на всіх масштабах.

Алгебраїчні фрактали базуються на використанні алгебраїчних функцій та властивостей.

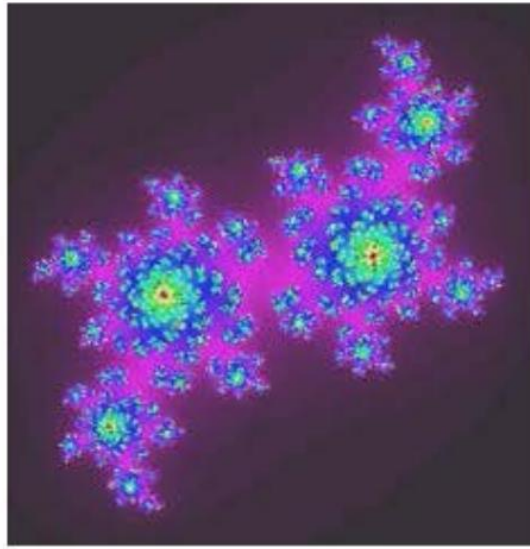


Рисунок 1.4 – Алгебраїчний фрактал

Стохастичні фрактали ґрунтуються на випадкових процесах та стохастичних розглядах.



Рисунок 1.5 – Стохастичний фрактал

Інша класифікація враховує природу походження фракталів, розрізняючи їх на рукотворні та природні.

Рукотворні фрактали створені людьми та визначаються фрактальними властивостями при будь-якому масштабі.

Природні фрактали обмежені максимальним і мінімальним розмірами, при яких спостерігаються фрактальні характеристики.

### 1.3 Огляд існуючих рішень

Розвиток методів та алгоритмів розпізнавання об'єктів на зображеннях стає дедалі актуальним рішенням у багатьох галузях життя. Тим самим змушує все більше і більше прогресувати та розвиватися дану технологію та усі супутні. Наразі кількість методів дозволяє обрати необхідний для тих чи інших потреб для досконалого результату.

Основними, універсальними та найпопулярнішими методами є:

1. нейронні мережі: використання штучних нейронних мереж дозволяє отримати доволі велику точність та має змогу до адаптація з різноманітними умовами;
2. шаблонний метод: в основі якого лежить метод порівняння об'єктів вже за попередньо відомими шаблонами;
3. Віюли Джонса: найчастіше використовується для розпізнавання обличчя, базується на каскадному класифікаторі;
4. контурний аналіз: базується на визначенні зовнішніх точок за допомогою контурів;
5. детектор точок: заснований на визначенні ключових точок та їх дескрипторів.

### Нейронні мережі

Нейронна мережа представляє собою високоякісну технологію, що використовує концепції, аналогічні роботі людського мозку. В основі людського мозку лежать нейрони, аналогічно штучній нейронній мережі, де основним елементом є штучний нейрон.

Штучний нейрон має певні синаптичні ваги на входах, що визначають силу вхідних зв'язків – менш важливі зв'язки мають меншу вагу і навпаки.

Обробка інформації відбувається через суматор та передаточну функцію або функцію активації.

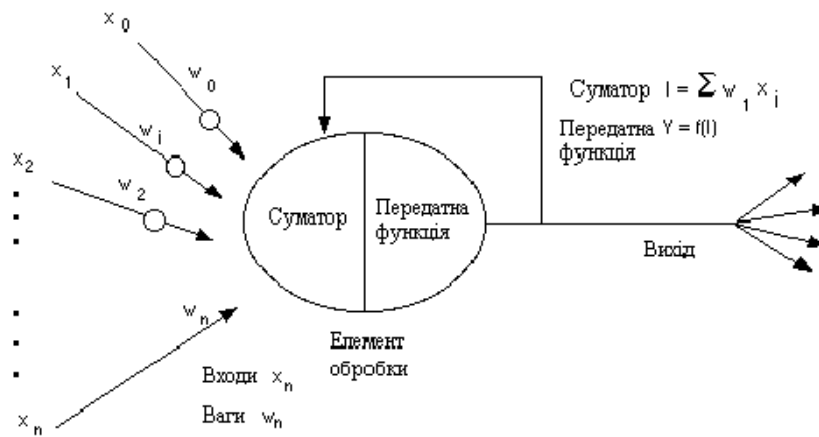


Рисунок 1.6 – Модель штучного нейрона

Існує доволі багато видів нейронних мереж, але основними є одношарові та багатошарові мережі.

Одношаровий перцептрон - це нейронна мережа, яка складається лише з одного шару нейронів. Вона має вхідний шар, безпосередньо з'єднаний з вихідним шаром, і використовується для вирішення простих задач класифікації або регресії. У кожному нейроні вхідного шару є ваги, які зважують вхідні сигнали, і суматор, який обчислює взважену суму вхідних сигналів. Результат подається на виході через функцію активації.

Одношарові перцептрони не можуть вирішувати складні завдання, які мають складну структуру даних чи вимагають виявлення внутрішніх залежностей. Вони є базовим елементом багатьох більш складних нейронних мереж, таких як багатошарові перцептрони.

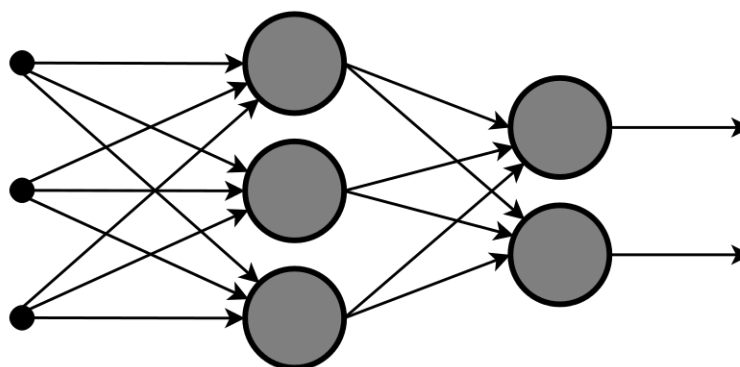


Рисунок 1.7 – Одношаровий перцептрон

Багатошаровий перцептрон (MLP) – це вид штучної нейронної мережі, що має кілька шарів нейронів, включаючи вхідний, приховані та вихідний шари. Кожен шар з'єднаний з попереднім і наступним шарами ваговими з'єднаннями. Кожне з'єднання



між нейронами має ваговий коефіцієнт, який визначає вагу з'єднання. Ці ваги навчаються під час тренування моделі.

Багатошарові перцептрони є потужними моделями, здатними навчатися складних завдань, таких як розпізнавання образів, мови, аналіз даних тощо. Тренування багатошарових перцептронів часто використовує алгоритми зворотнього поширення помилки та градієнтного спуску. Багатошаровий перцептрон є ключовим елементом глибокого навчання і представляє собою гнучку архітектуру для різноманітних завдань машинного навчання.

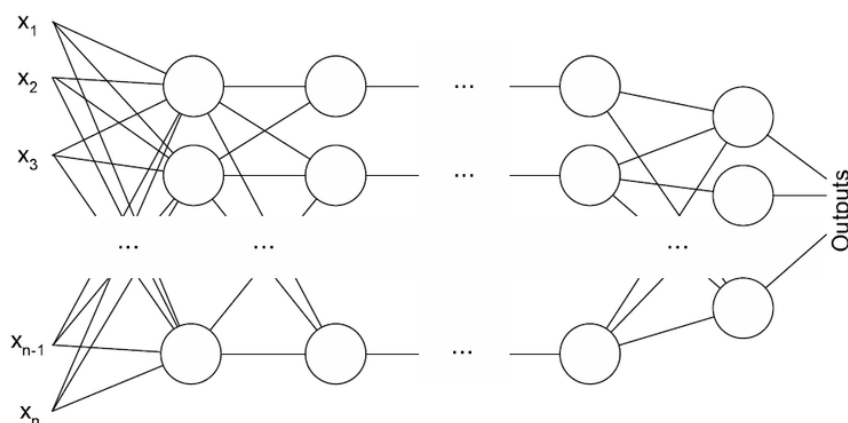


Рисунок 1.8 – Багатошаровий перцептрон

### Шаблонний метод

Шаблонний метод розпізнавання об'єктів заснований на однакових за своєю структурою та суттю шаблонах. Алгоритм його роботи доволі простий, він спирається у порівнянні поточного об'єкту із ідеальним за своєю структурою шаблоном. Найбільш популярний при розпізнаванні тексту, так званий OCR.

Шаблонні методи проявляють високу стійкість до дефектів на зображеннях і забезпечують швидку обробку вхідних даних. Однак їх ефективність обмежена тим, що вони ефективно розпізнають лише ті шрифти або шаблони, які вже включені в їх базу даних. Якщо розпізнаваний шрифт навіть трохи відрізняється від відомого еталону, шаблонні методи можуть допускати помилки, особливо при обробці високоякісних зображень. Існують певні види розпізнавання об'єктів за шаблонним методом:

1. проста відповідність - це основний метод порівняння двох зображень. Він ґрунтується на тому, щоб визначити, наскільки добре два зображення співпадають

або відрізняються. Цей метод може використовувати такі метрики, як сума квадратів відхилень (SSD), середнє квадратичне відхилення (MSE) або інші показники відповідності між пікселями зображень;

2. відповідності за характерними рисами - це метод, що визначає відповідності між характерними рисами або ключовими точками на двох зображеннях. Для цього можуть використовуватися детектори ключових точок, такі як SIFT або SURF. Відповідність робиться на основі співставлення описів цих ключових точок;

3. відповідність за областю – це метод в якому відповідність порівнюється для цілих областей зображень. Методи, такі як гістограма орієнтованих градієнтів (HOG) чи методи, засновані на локальних бінарних шаблонах (LBP), можуть використовуватися для опису та порівняння областей;

4. кореляція зображень це метод, що визначає ступінь взаємодії між пікселями двох зображень. Він враховує ступінь схожості між яскравістю або кольорними властивостями відповідних пікселів. Висока кореляція вказує на схожість, тоді як низька кореляція вказує на відмінність.

### **Метод Віола Джонса**

Незважаючи на стрімкий розвиток та популярність нейронних мереж, метод Віоли Джонса залишається на провідних місцях у галузі розпізнавання об'єктів. Але насамперед цей метод є особливо ефективним при роботі з людським обличчям та відеоматеріалах, що відтворюються у реальному часу. До базових принципів на які спирається метод Віоли Джонса є:

1. інтегральне представлення зображення: Використання матриці інтегрального представлення зображення, розмірність якої відповідає піксельній розмірності оригінального зображення. Це прискорює визначення областей інтересу та об'єктів.

2. використання ознак Хаара: Визначення потрібного об'єкта за його характеристиками, використовуючи ознаки Хаара, які є ефективними для виділення особливостей на зображенні.

3. бустинг: Застосування алгоритму підсилення (бустингу), який покращує якість визначення ознак об'єкта. Цей алгоритм дозволяє поєднувати слабкі класифікатори для створення потужного класифікатора.

4. передача всіх ознак класифікатору: Ознаки Хаара передаються на вхід класифікатору, який видає результат "істина" або "хиба". Це спрощує процес прийняття рішення.

5. додатковий каскад ознак: Використання додаткового каскаду ознак дозволяє відкидати ті частини зображення (вікна), в яких не виявлено об'єкта, що допомагає прискорити визначення регіонів інтересу та підвищити точність розпізнавання.

### **Контурний аналіз**

Контурний аналіз - це один з методів обробки та аналізу зображень, який перш за все спирається на контур та ігнорує внутрішні точки зображення. Головною ідеєю цього методу є виділення границі об'єктів та аналізувати їх форму, розташування, тощо. Алгоритм виконання контурного аналізу має в собі три головних етапи, а саме:

1. знаходження та виділення контурів об'єктів. В основі цього етапу лежить обробка контурів об'єктів за рахунок виявлення переходів між областями з різними графічними характеристиками, які виконуються за допомогою градієнтів, фільтрів, математичних чи оптичних алгоритмів;

2. відокремлення знайдених об'єктів контурними лініями, які були отримані під час першого етапу. Для формування контурів використовуються спеціалізовані алгоритми відокремлення та замкнення;

3. аналіз контурів здійснюється за допомогою вивчення характеристик контурів, за допомогою набутих характеристик (довжина, вигини, орієнтація, форма, тощо ). Згодом на базі цих властивостей виконується класифікація об'єктів на зображенні.

Контурний аналіз надає можливість описувати, зберігати, порівнювати і розпізнавати об'єкти, представлені своїми зовнішніми контурами. Цей метод

допомагає вирішувати ключові завдання розпізнавання образів, такі як трансформації, повороти та зміна масштабу зображення об'єктів.

### Детектор точок

Детектор точок – це алгоритм обробки зображень, який спеціалізується на визначенні та локалізації ключових точок чи особливих точок в зображенні. Особливі точки можуть мати унікальні властивості, такі як велика зміна яскравості, розгалуження або інші характеристики, які роблять їх легко виявляємими та відслідковуваними в різних умовах.

Детектори точок часто використовуються в задачах розпізнавання образів, обробки зображень та комп'ютерного зору для розв'язання завдань, таких як вирізання об'єктів, слідкування за рухом та визначення ключових особливостей в зображеннях. З часом використання було виділено основні вимоги до особливих точок задля найбільшої точності та ефективності розпізнавання об'єкту, а саме: відмінність, стабільність, локальність, точність, кількість, інваріативність, інтерпретованість.

Найбільш популярним детектором є детектор Харріса. В основі алгоритму є:

1. обчислення градієнту зображення в кожному пікселі з використанням гаусового розмиття;
2. обчислення матриці других моментів  $M$  для вікна навколо кожного пікселя;
3. обчислення відгуку кута;
4. відсічення за порогом;
5. знаходження локальних максимумів функції відгуку у визначеному радіусі;
6. визначення найсильніших локальних максимумів.

Формула обчислення матриці Харріса (1.1):

$$M = \sum_{(u,v) \in W} W(u,v) \begin{bmatrix} l_x^2 & l_x l_y \\ l_x l_y & l_y^2 \end{bmatrix} \quad (1.1)$$

де  $W(u,v)$  – вагова функція;

$M$  – автокореляційна матриця.

Формула обчислення кута відгуку (1.2):

$$R = \det M - k(\text{tr} M)^2 > k \quad (1.2)$$

де  $R$  – міра кута;

$k$  – емпірична константа.

Застосування детектора Харріса до зображення.



Рисунок 1.9 – Результат детектора Харріса

Алгоритм детектора Харріса допомагає вирізняти ключові точки на зображеннях, що є корисним у багатьох задачах, включаючи відтворення об'єктів та визначення їхніх характеристик.

#### 1.4 Специфікація вимог програмного забезпечення

Визначення специфікація є важливим етапом проектування та моделювання будь-якого програмного забезпечення. Для успішного складання опису вимог використовують сценарії розробки. У сценаріях перш за все відображаються різноманітні варіанти використання програмного забезпечення.

Першочерговим завданням при розробці програмного забезпечення є визначення його головне призначення, а вже після цього складати загальний опис функціональних та нефункціональних вимог.

Спираючись на вище сказане, слід визначити, що головним призначенням, є розпізнавання та сегментація різноманітних об'єктів на зображеннях, використовуючи при цьому методи для розпізнавання та сегментації. Виходячи з цього можна зробити висновок, що уся архітектура майбутнього програмного забезпечення повністю побудована тих самих методах. То визначення функціональних вимог є доволі обмеженим.

До них відносяться наступні елементи:

1. завантаження необхідного зображення до програмного забезпечення;
2. початок обробки зображення, шляхом визначення поточного завдання, а

саме вибір між розпізнаванням чи сегментацією об'єктів;

### 3. вивантаження обробленого зображення.

До інтерфейсних вимог відносяться елементи взаємодії користувача з інтерфейсом програми. Інтерфейс має бути інтуїтивно зрозумілий та мати усі необхідні засоби задля виконання функціональних вимог програмного забезпечення.

Безпека є доволі важливим аспектом сьогодення, саме тому безпекові вимоги є базовими та вимагають уваги. Розроблювальне програмне забезпечення є доволі безпечним з боку зберігання оброблених зображень, оскільки зберігають усе на внутрішніх жорстких дисках.

## **Висновки до розділу 1**

Проаналізовано предметну сферу розпізнавання, сегментації та обробки зображень. Знання у цій сфері надають повне розуміння усієї картини. Чітке уявлення про методи та алгоритми дає можливість розробки спеціалізованого програмного забезпечення.

Крім того, досліджено основні методи обробки, які дають змогу та можливість орієнтуватися у використовуваних алгоритмах для тих чи інших умов.

Також більш детально розглянуто вже існуючі рішення та визначено головну проблематику розпізнавання та обробки зображення.

Визначено специфікації вимог до програмного забезпечення.

## 2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

Для успішного розпізнавання об'єктів на зображеннях застосовують різноманітні методи. Але першочергово для цифрової обробки виконується локалізація областей зображення задля визначення об'єкту на зображеннях в залежності від умов та характеристик.

У результаті виконання локалізації стоять координати об'єктів за якими будують межі задля візуалізація розташування об'єкта.

Існує певні базові методи задля вдалого відокремлення областей на зображеннях. Класифікація методів містить у собі 4 метода, що зображені на рисунку 2.1.

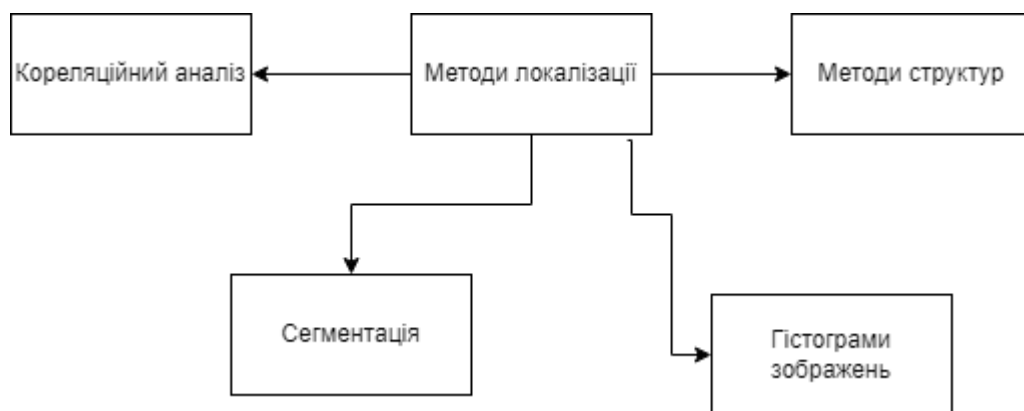


Рисунок 2.1 – Класифікація методів

Задля розробки програмного забезпечення було обрано метод сегментації.

Сегментація дозволяє ефективно розбити зображення на окремі сегменти, кожен з яких має власні характеристики та властивості, за рахунок цього відбувається відокремлення об'єкту від фону за контурами.

Також слід зазначити, що задля ефективного розпізнавання та отримання бажаного результату, необхідно зробити певні операції, а саме: згладжування, а також застосування одного з алгоритмів розпізнавання об'єктів, що були розглянуті у першому розділі КРМ.

## 2.1 Математична модель сегментації зображень

Сегментація – це невід'ємна частина процесу цифрової обробки, головною задачею якої є поділ зображення на певні області. Деталізація визначається безпосередньо у процесі виконання, коли необхідні області чи об'єкти відокремлені для подальшого аналізу.

Під час аналізу предметної сфери були визначені основні алгоритми сегментації зображень, а саме:

1. порогова сегментація;
2. на основі країв;
3. на основі регіонів;
4. на основі кластеризації;
5. на основі нейронних мереж.

Проаналізуємо деякі з цих методів та сформуємо їх математичну складову.

### Порогова сегментація

Сегментація за пороговим значенням – це базовий метод виконання сегментації зображень, частіше за все використовується для відокремлення об'єктів від фону. Об'єкти виділяються у відтінках сірого.

Під час реалізації для кожного з пікселів визначається його інтенсивність на рівні сірого кольору, але якщо набута інтенсивність перевищує певний встановлений поріг, то автоматично пікселю прирівнюється максимальне значення 255, що є відповідником білого кольору. У протилежному випадку встановлюється 0, що є чорним кольором. Таким чином області зображення на яких інтенсивність сірого вище отримують білий колір, а усі інші – чорний.

У формулі (2.1) записана математична модель алгоритму.

$$g(x, y) = \begin{cases} 255, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases}, \in [0, 255] \quad (2.1)$$

Оскільки порогова сегментація є базовим методом, то можна виокремити дві групи методів, що базуються на ній, а саме: метод локального та метод глобального порогу.



Методи локального порогу – це методи які використовуються різноманітні порогові значення до різних областей. Значення цих порогів отримується на основі сусідніх пікселів. Це надає змогу обробляти зображення з різними ступенями контрастності та яскравості. У свою чергу цей метод має ряд певних базових алгоритмів, а саме: Бернсена, Ніблека та Савуолі.

Методи глобального порогу – це методи, що роблять розподіл зображення на дві області (фон та об'єкт). Однак для роботи та реалізації цього методу необхідно використовувати попередньо заданий поріг. Задля його визначення було винайдено певні алгоритми, що базуються на різноманітних підходах. До таких методів відносяться: метод Оцу, ітеративний метод, гістограмний метод.



Рисунок 2.2 – Порогова сегментація

### Метод на основі країв

Метод в основі якого є виділення контуру на зображеннях. Він використовує сплайни.

Сплайн – це певний математичний вираз набору багаточленів задля відображення геометричних фігур, наприклад кривих. Цей метод не є доволі ефективним оскільки вимагає попередньої умови щодо бажаної форми. Сплайни обраховуються за формулою визначення параметричної кривої (2.2).

$$c(s, t) = \begin{cases} x(s, t) \\ y(s, t) \end{cases}, s \in [0, 1], t \in [0, \infty], \quad (2.2)$$

де  $x$  та  $y$  – координати двомірної кривої,

$s$  – лінійний параметр,

$t$  – номер параметру.

На рисунку 2.3 зображена ідея використання методу на основі країв

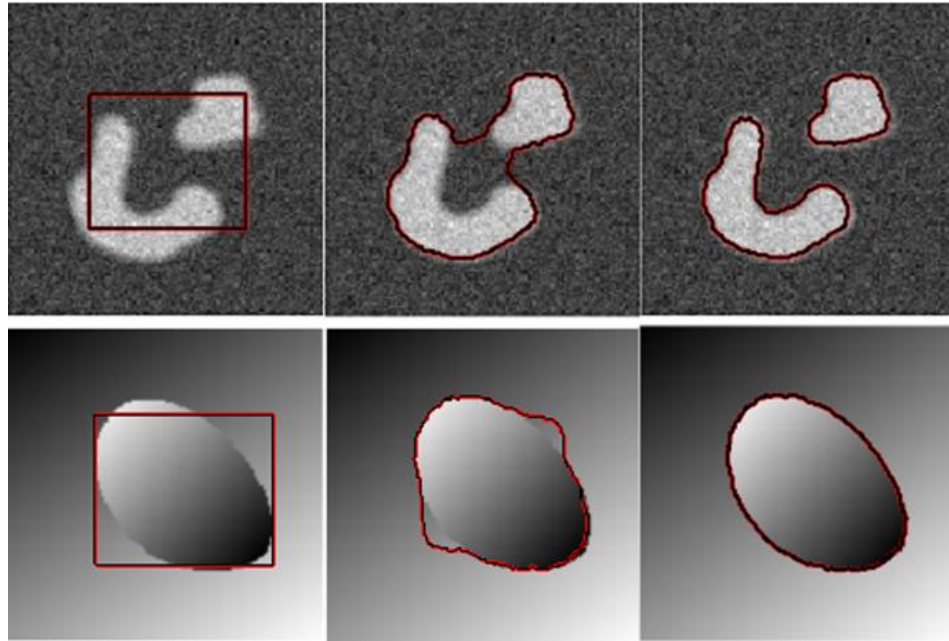


Рисунок 2.3 – Використання методу на основі країв

Найчастіше цей метод використовується в медичній практиці, а також при обробці зображень, де досить важливим є локалізація та точність визначення об'єктів.

### **Метод на основі кластеризації**

Кластеризація - це метод який базується на основі окремих груп у просторі. Головною задачею цього методу є розподіл набору даних на кластери, таким чином, щоб усі дані, що містяться у межах одного кластера були найбільш схожі. Кожен з кластерів має свою особливу форму та структуру, за рахунок чого можливо легко виділити ту чи іншу область.

Метод кластеризації є одним з найпопулярніших методів та знайшов застосування у багатьох галузях. Головною особливістю є взаємодія з кольорами пікселів, що надає можливість більш детально обробляти кольорові зображення. У метод закладено 2 різноманітних алгоритми а саме: K-means, C-means.

Метод K-means – один за найпопулярніших методів реалізації методу кластеризації. Він є доволі ефективним через власну швидкою та точність сегментації. В основі цього алгоритму є розподіл даних на певну кількість K кластерів, які базуються на K центроїдах, що є центром тих самих кластерів. Ефективне розміщення центроїдів у просторі є запорукою успіху виконання

алгоритму. Кожен об'єкт, що містить вектори повинен бути призначений певному кластеру з множини. Головною метою є мінімізація суми квадратів відхилень між встановленими точками та центроїдами. Математична модель виконання мети має наступне рішення (2.3)

$$\operatorname{argmin}_S \sum_{i=1}^K \sum_{x \in S_i} \|x - m_i\|^2, \quad (2.3)$$

Де  $m_i$  – середнє значення векторів у кластері

Алгоритм є ітеративним, за рахунок чого підвищується точність результату.

Для успішного виконання треба виконати два кроки

1. призначити кластери векторам;

$$S_i^{(t)} = \left\{ x_p : \|x_p - m_i^{(t)}\|_2^2 \leq \|x_p - m_j^{(t)}\|_2^2 \quad \forall j, 1 \leq j \leq K \right\}, \quad (2.4)$$

Де вектор  $x_p$  має відношення тільки до одного з кластерів  $S_i^{(t)}$

2. оновити центроїди.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x \in S_i^{(t)}} x \quad (2.5)$$

Кінцевим результатом роботи алгоритму є коли призначення кластеру вектору збігається і не дає нового результату.

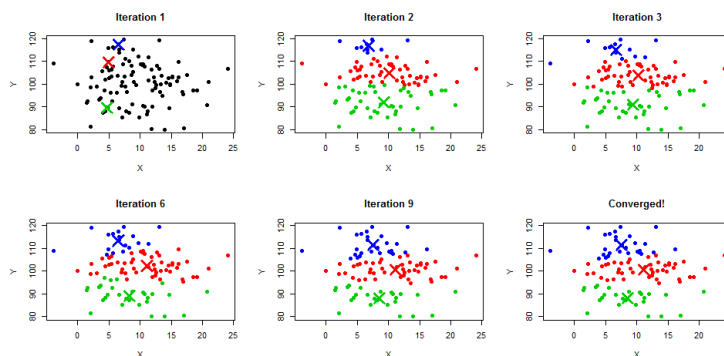


Рисунок 2.4 – Метод K-means

Недоліком є повна залежність в першочергово обраних центроїдів.

Метод C-means – це алгоритм який є покращення алгоритму K-means, оскільки він надає можливість розрахунку ступеня належності елемента до того чи іншого кластеру. Основі кроки його виконання:

1. визначення центроїдів;

2. розрахунок матриці належності до певного кластеру;

$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}}, \quad (2.6)$$

Де  $d_{ij}$  – це відстань між  $i$  вектором та  $j$  центроїдом, параметр  $m \in [1, \infty]$  контролює нечіткість кластерів, чим більше тим нечіткіше.

3. обчислення центроїдів;

$$m_j = \frac{\sum_{i=1}^n \mu_{ij}^m x_i}{\sum_{i=1}^n \mu_{ij}^m}, \quad \forall j = \overline{1, C} \quad (2.7)$$

4. ітерувати попередні кроки, поки не буде виконуватися умова, що функція втрат зменшується після кожної ітерації.

$$J = \sum_{i=1}^n \sum_{j=1}^C \mu_{ij}^m d_{ij} \quad (2.8)$$

### Метод на основі згорткових нейронних мереж

Згорткова нейронна мережа – це один з типів нейронних мереж, який спеціально розроблений для роботи зі структурованими сітковими даними. Головною ідеєю є автоматичне дослідження ієрархії властивостей та особливостей та сортувати їх від простих до складних.

У математичному аспекті згорткові шари використовуються задля виявлення локальних особливостей в різних областях зображення. Математична модель згорткової нейронної мережі містить у собі такі елементи як: згорткові шари, шари для підсумування, повнов'язні шари.

Згортковий шар використовує в своїй основі спеціальні ядра чи фільтри задля проведення операцій над вхідними даними. Операція згортки розраховується за наступною формулою.

$$(X * K)_{ij} = \sum_{m,n} X_{i+m,j+n} * K_{m,n}, \quad (2.9)$$

Де,  $X$  – вхід в матрицю,

$K$  – ядро матриці.

Шар підсумовування – це шар який використовується для фільтрації та зменшення розмірів вихідних даних. У базі є операція максимальної вибірки (2.10), де визначається максимальне значення області.

$$Y_{ij} = \max_{m,n} X_{i*s+s,j*s+n}, \quad (2.10)$$

Де,  $Y_{ij}$  – це результат операції максимальної вибірки,

$S$  – розмір області

Повнов'язний шар – це шар, що поєднує всі виходи попередніх шарів зі штучним нейроном поточного шару. Математично обчислюється за наступною формулою(2.11)

$$Z_i = f(\sum_j W_{i,j} * Y_j + b_i), \quad (2.11)$$

Де,  $W_{i,j}$  – вага між штучними нейронами,

$Y_j$  – вихід з попереднього шару,

$b_i$  – зміщення,

$f$  – функція активації.

## 2.2 Методи згладжування зображення

Згладжування зображення використовується задля збільшення точності та деталізації отриманого зображення. Згладжування значно спрощує розпізнавання, оскільки за рахунок нього прибирається багато шумів та зернистості.

Один з головних методів згладжування є метод Гауса. Він використовує Гаусів фільтр за рахунок якого зменшується контрастність та виділяються компоненти з високою частотою. У базі фільтра Гауса лежить зображення з ядром Гаусса. Ядро – це двовимірна функція, що застосовується для розрахунку ваги, яка застосовується до кожного з пікселів. Ядро обчислюється за формулою (2.12) Для кожного пікселя зображення обчислюється нове значення, в основі якого лежить середнє вагове його сусідів.

$$G(x, y) = \left( \frac{1}{2\pi * \sigma^2} \right) * e^{\left( -\frac{(x^2+y^2)}{2*\sigma^2} \right)}, \quad (2.12)$$

Де  $x$  та  $y$  – відстань від центру ядра,

$\sigma$  – це ступінь сігма, ступінь розмиття.



Рисунок 2.5 – Згладжування Гаусса

Найбільш ефективно використовується для розм'якшення контурів та видалення шуму. Поєднання з іншими методами згладжування поліпшує ефект.

### **2.3 Архітектури згорткових нейронних мереж**

Розглядаючи усі можливі архітектури розроблювального програмного забезпечення можливо зробити висновок, що кожна з них має свої переваги та недоліки.

До найпопулярніших методів та алгоритмів, що стосуються розпізнавання є наступні: HOG, R-CNN, SSD, YOLO, RetinaNet, тощо.

#### **HOG**

Histogram of Oriented Gradients (HOG) - це метод для визначення об'єктів на зображеннях, основний принцип якого полягає в аналізі градієнтів яскравості в зображенні.

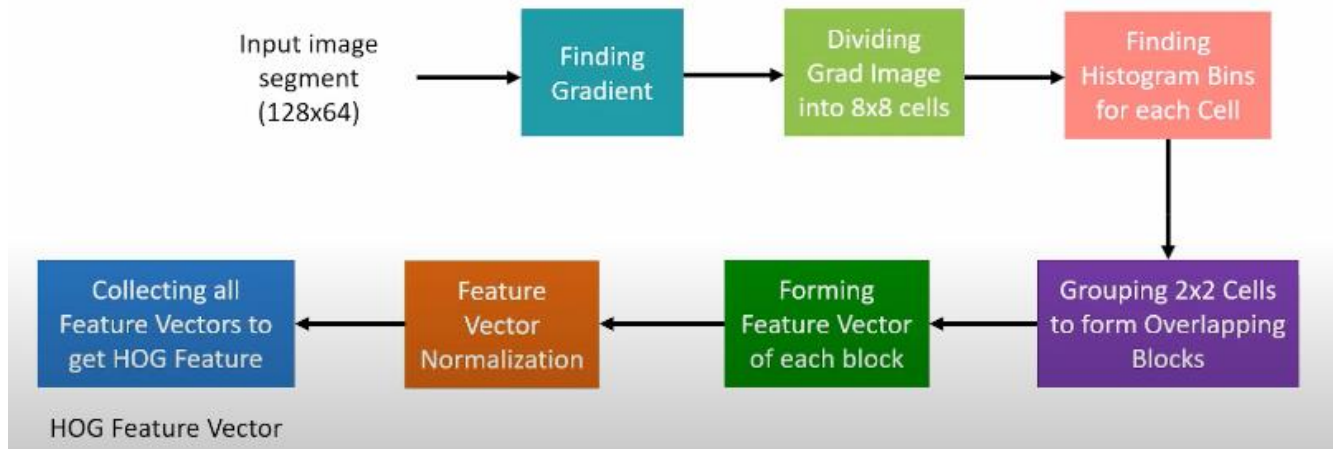


Рисунок 2.6 – Архітектура HOG

Основні етапи архітектури HOG:

1. зображення нормалізується та вирівнюється, щоб поліпшити якість обробки та уникнути впливу освітлення;
2. визначається градієнт для кожного пікселя зображення, що може бути виконано за допомогою фільтрів;
3. зображення розділяється на невеликі клітини, кожна з яких представляє собою невеликий регіон;
4. обчислення гістограм орієнтації, відображаючи розподіл напрямків градієнтів у цій клітині;
5. групування клітин в блоки, і об'єднання гістограм орієнтації клітин в цьому блоці;
6. нормалізація векторів гістограм орієнтацій у кожному блоку для компенсації змін яскравості та поліпшення стійкості;
7. нормалізовані блоки об'єднуються в один вектор, який служить описом об'єкта для подальшого використання в завданнях, таких як класифікація.

Архітектура HOG дозволяє виявлення об'єктів за їх текстурою та формою, що робить його ефективним для завдань об'єктного визначення в зображеннях.

## R-CNN

R-CNN – це одна з архітектур глибокого машинного навчання для визначення об'єктів на зображеннях. Архітектура використовує згорткові нейронні мережі задля обробки зображень.



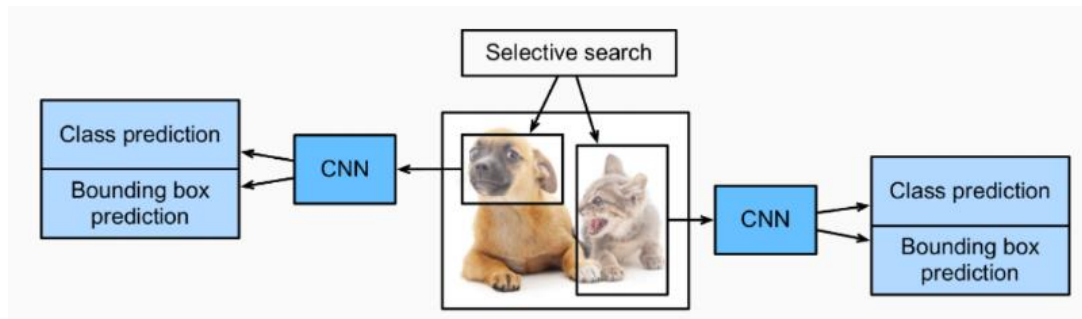


Рисунок 2.7 – Архітектура R-CNN

Основні етапи архітектури R-CNN:

1. застосування методу Selective Search, для визначення теоретичних областей які містять об'єкти;
2. використання згорткової нейронної мережі, задля обробки заздалегіть визначених областей та формування вектора ознак;
3. класифікація об'єкта за допомогою визначеного вектора, а також корегування областей здаля точності кінцевого результату;
4. об'єднання пропозицій;
5. кінцеві результати, що містять класифікацію об'єкта та його координати охоплення.

## SSD

SSD – це архітектура для визначення об'єктів на зображеннях. Головною перевагою є швидкодія та здатність розпізнавати об'єкти різних класів на різних рівнях роздільної здатності.

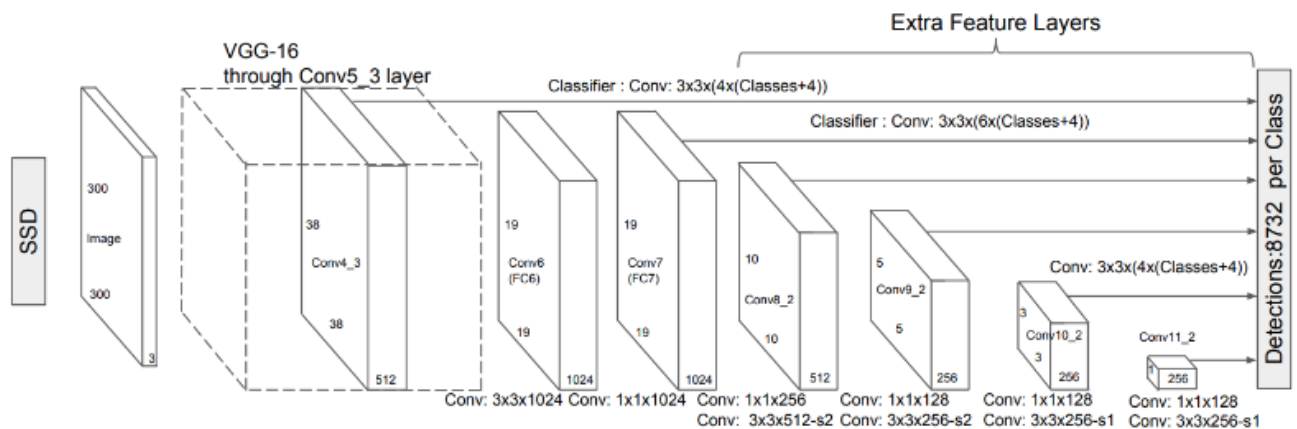


Рисунок 2.8 – Архітектура SSD



## Основні етапи архітектури SSD:

1. генерація особливостей для різних рівнів роздільної здатності;
2. використання детекторів для кожного рівня;
3. виведення детекторів з прогнозами для кількох класів;
4. відсіювання за допомогою NMS для усунення дублювання та покращення точності результату.

## YOLO

YOLO чи You Only Look Once – це новітній метод задля визначення та локалізації об'єктів на зображеннях. За рахунок розвитку алгоритмів має головну перевагу над усіма іншими, а саме можливість одночасно визначати та локалізувати об'єкти з високою швидкістю обробки.

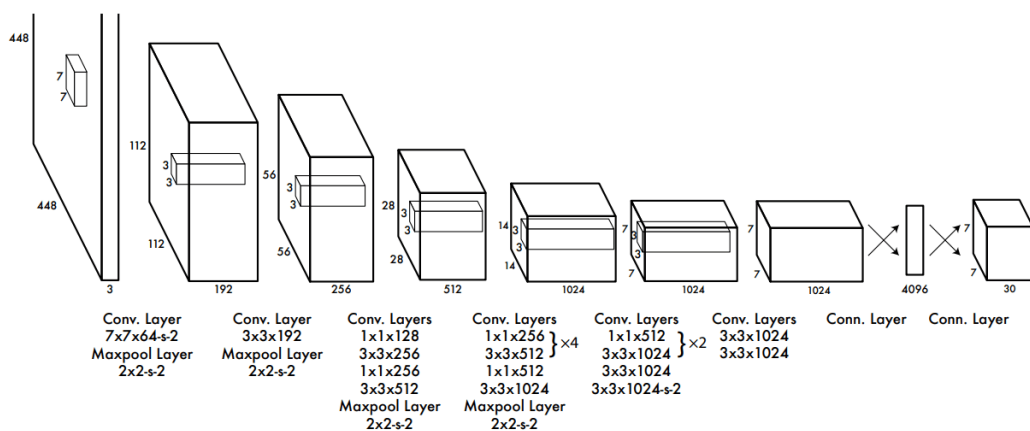


Рисунок 2.9 – Архітектура YOLO

## Основні етапи архітектури YOLO:

1. подача зображення до нейронної мережі
2. витягування важливих аспектів з допомогою згорткової нейронної мережі;
3. згладжування зображення задля прибирання зайвих деталей та шумів;
4. кожен елемент сформованої сітки отримує bounding box з параметрами, а саме координатами, розміром та ймовірністю задля кожного об'єкту класу;
5. генерація виходу та інформації про розміщення об'єктів на зображенні;
6. відкидання малих ймовірностей.

За рахунок такої будови YOLO може розпізнавати елементи лише за одну ітерацію, що забезпечує ефективну та швидку роботу.

## 2.4 Порівняння методів згорткових нейронних мереж

Машинне навчання – це галузь ШІ, яка виконує завдання навчання комп'ютера задля підвищення ефективності у будь-яких задачах. Завдання розпізнавання об'єктів не є винятком та потребує навчання.

Головними етапами навчання згорткових нейронних мереж є:

1. подання вибірки зображень, які треба буде визначити;
2. визначення ознак задля класифікації об'єктів та визначення класів;
3. побудова гіпотез та визначення місцезнаходження самого об'єкта.

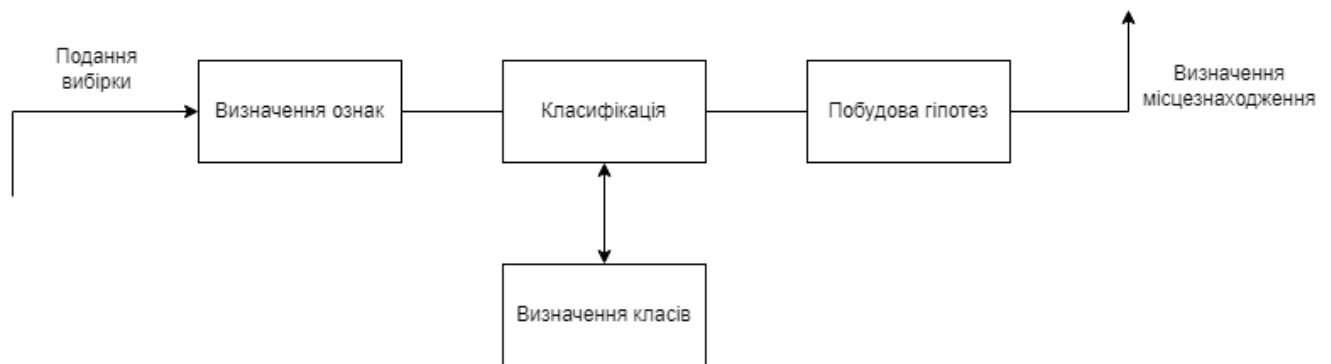


Рисунок 2.10 – Етапи навчання нейронної мережі

Залежність між коефіцієнтом та точністю визначення класу об'єкту пряма. Оскільки чим більше коефіцієнт, тим більше вірогідність визначення об'єктів, однак згорткові нейронні мережі не завжди можуть вірно визначити тип об'єкту, у зв'язку з чим виникає зменшення продуктивності.

Задля визначення найбільш відповідного методу розпізнавання об'єктів було проведено дослідження швидкодії та точності. Була використана вибірка з понад ніж 700 різноманітних класифікаторів. Частина з цього датасету була використана задля навчання згорткової нейронної мережі, інша для дослідження.

Тести були розбиті на певні проміжки, а саме 10, 100 та 350 ітерацій.

Таблиця 2.1 – Некоректність мереж

Назва	10	100	350
HOG	70%	18%	3%
R-CNN	63%	22%	3.2%
SSD	65%	17%	3.3%
YOLO	59%	32%	2%

Аналізуючи отримані дані слід зазначити, що при 10 ітераціях похибка визначення є доволі значною більше ніж 50%, але це пояснюється тим, що нейронна мережа тільки наповнює базу класів. З кількістю ітерацій точність визначення зростала та у кінцевому результаті складає лише декілька процентів, що є довільним результатом. Найкращий показник показав метод YOLO.

Після завершення навчання нейронних мереж було проведено остаточне тестування задля визначення базових результатів за точністю та швидкодією.

Таблиця 2.2 – Точність та швидкодія

Назва	Точність	Швидкодія
HOG	25%	0.5сек
R-CNN	55%	1.6сек
SSD	76%	0.7сек
YOLO	65%	0.2сек

Аналізуючи отримані дані слід зазначити, що найбільшою точністю відзначився метод SSD, але значно поступився швидкодією. Метод R-CNN показав найгірші результати зі швидкодії, а метод HOG з точності. Підбиваючи підсумки було визначено, що алгоритм YOLO є найбільш ефективним та вирішено використати у реалізація програмного забезпечення розпізнавання та сегментації об'єктів. Алгоритм

YOLO має такі особливості як:

1. швидкодія;
2. один прохід мережі;
3. відкритий код;
4. має різні версії розробки;
5. точність.

Існують різноманітні версії алгоритму YOLO, а саме: YOLOv1, YOLOv2, YOLOv3. Кожна з цих версій доповнює наступну, на даний час YOLOv3 є останньою версією та містить у собі найбільш ефективні рішення, що суттєво впливають на швидкодію та точність розпізнавання.

## **2.5 Розробка діаграми використання**

Діаграма використання, або діаграма використання прецедентів, є одним з інструментів, що використовується для моделювання функціональності системи з точки зору її взаємодії з зовнішніми агентами (користувачами, іншими системами або компонентами).

Після детального огляду методів, алгоритмів сегментації та розпізнавання зображень, а також аналізу згорткових нейронних мереж та їх архітектур, було розроблено діаграму використання застосунку.

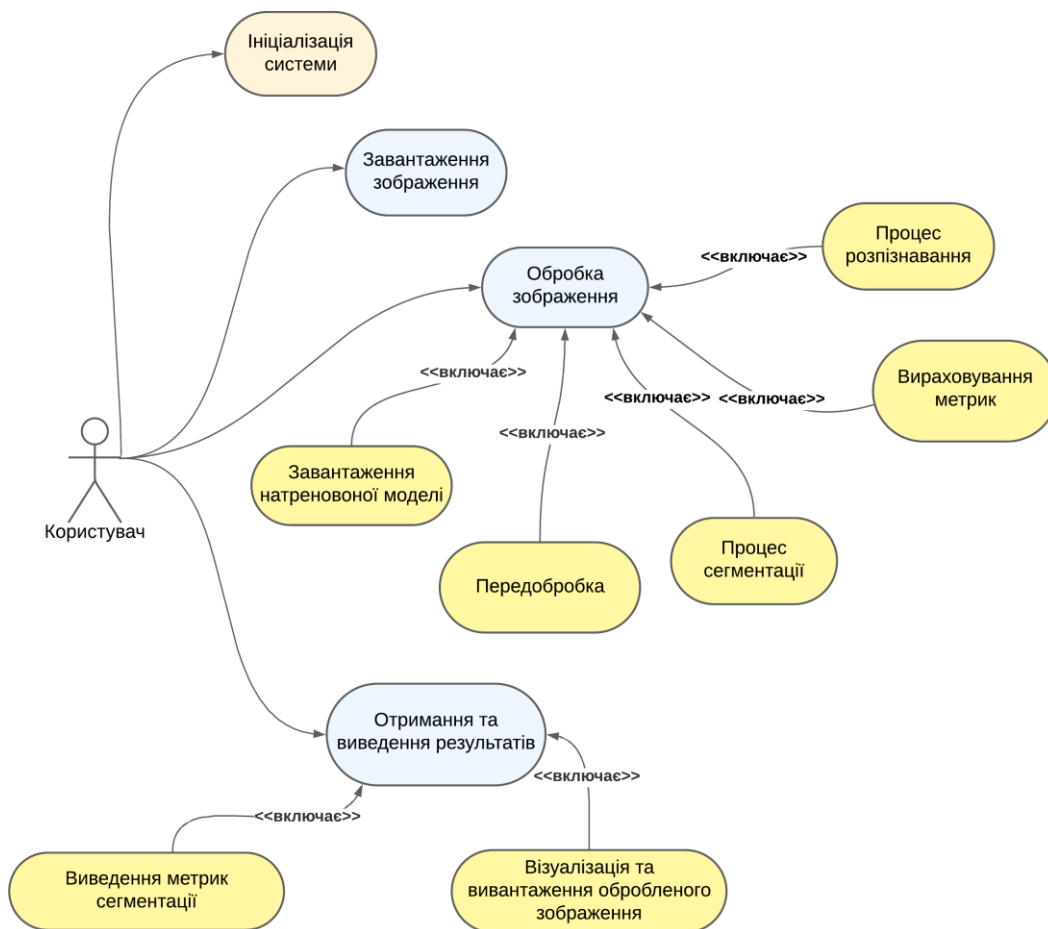


Рисунок 2.11 – Діаграма використання застосунку

Розглядаючи діаграму використання, можна зазначити, що найскладнішим та продуктивним етапом є етап аналізу та обробки зображення. Враховуючи можливі поліпшення застосунку та імплементація нових методів кількість сценаріїв та етапів використання може змінюватися так само.

Розглядаючи надану діаграму можна визначити її морфологічну структуру застосунку. Під час запуску застосунку система ініціалізується та оброблює усі необхідні бібліотеки та модулі. Далі користувач завантажує необхідне зображення, після цього система оброблює це зображення, проходячи усі необхідні етапи починаючи від обробки натренованої моделі та закінчуючи визначенням метрик. Після обробки зображення виводиться на екран, а також вивантажується у спеціальне місце.

## Висновок до розділу 2

Визначено та класифіковано методи для успішного розпізнавання та сегментації об'єктів на зображеннях. Розглянуто предмет сегментації та досліджено основні методи, а саме:

1. метод порогової сегментації;
2. на основі країв;
3. на основі кластеризації;
4. метод згорткових нейронних мереж.

Для кожного з цих методів, визначено математичні моделі, що дозволяє більш глибоко визначити предмет роботи.

Також розглянуто оператори згладжування зображення, визначено їх необхідність та математичну сторону.

Досліджено архітектури згорткових нейронних мереж, визначені основні етапи кожного з алгоритмів та за рахунок цього проведено дослідження у якому визначалась ефективність алгоритмів згорткових нейронних мереж за двома критеріями, а саме: точність та швидкодія. У результаті було обрано алгоритм YOLO, після чого визначено його особливості.

Розроблено діаграму використання, а також розглянуто морфологічну структуру застосунку.

## **3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ**

### **3.1 Програмні засоби для розпізнавання, сегментації зображень**

Для успішної розробки будь-якого застосунку є дуже важливим аспектом вибір програмних засобів. У контексті цього застосунку можна зазначити, що до таких засобів відносяться: бібліотеки, мови програмування та середовища розробки.

#### **Python**

В ході розробки застосунку задля розпізнавання та сегментації зображень було використано мову програмування Python. Використання цієї мови має багато переваг особливо задля галузі машинного навчання та обробки зображень. Велика кількість супутніх бібліотек дозволяють доволі просто реалізовувати алгоритми та методи розпізнавання та сегментації, а також обробляти та імпортувати вже навчені моделі.

Python доволі сучасна мова програмування, то ж має простий та ємкий синтаксис, що позитивно впливає на розробку застосунку. Варто сказати, що можливість кросплатформної розробки є також важливим аспектом, оскільки тематика застосунку є доволі сучасною темою та буде досліджуватися й надалі, то ж нема гарантії на якій основі в майбутньому буде операційна система пристрою на якій буде проводитися дослідження. Це може бути як Windows, так і Linux чи macOS.

Маючи велике ком'юніті відкривається доступ до багатої документаційної бази, а також форумів, підручників, тощо. Це також впливає на швидкість та якість розробки застосунку.

То ж у висновку можна зазначити головні переваги обраної мови, а саме до них відносяться наступні пункти:

1. велика кількість бібліотек;
2. зрозумілий синтаксис;
3. кросплатформність;
4. велика спільнота розробників.

## PyCharm

Задля розробки було використано IDE PyCharm, яке перш за все орієнтоване на роботу з мовою програмування Python. Інтегроване середовище розробки містить у собі багато зручних та корисних інструментів, які значно оптимізують розробку. Насамперед автодоповнення коду, яке у свої базі містить аналіз вже написаного коду. Також середовище має великий спектр інструментів для рефакторінгу, які можуть як змінювати та вдосконалювати код задля читабельності.

Вбудовані інструменти для роботи з базою даних, з системою контролю версій, бібліотеками та методами машинного навчання є головними перевагами PyCharm.

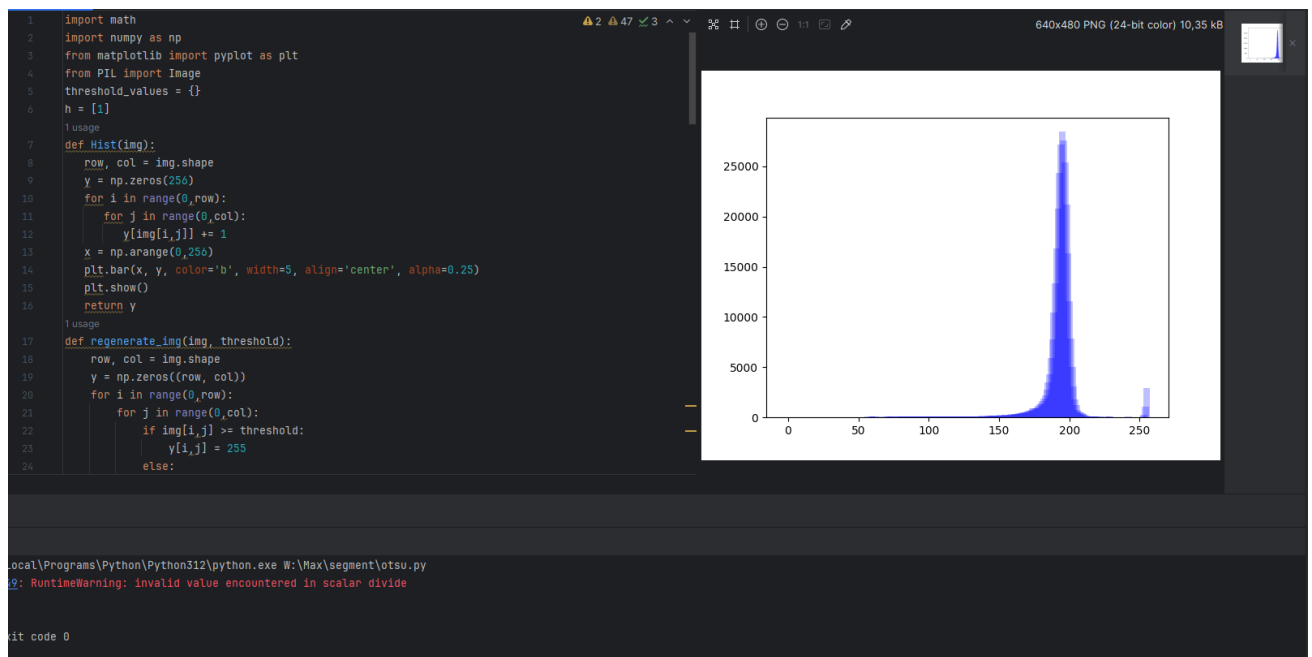


Рисунок 3.1 – Інтерфейс PyCharm

Віртуальне середовище та інтерпретатори значно полегшують розробку та тестування застосунків на мові Python. Усі наведені вище аспекти роблять це середовище одним з найкращих задля професійної розробки.

## Огляд бібліотек для обробки даних

Використання бібліотек є головним аспектом мови Python, їх застосування значно пришвидшує та оптимізує розробку застосунків. Задля машинного навчання існує велика кількість необхідних бібліотек, які допомагають навчити ті чи інші моделі, впроваджувати необхідні методи та алгоритми, проводити статистичний



аналіз за допомогою спеціалізованих бібліотек для побудови графіків. Сукупне застосування бібліотек сприяє якісній обробці даних.

### **NumPy (Numerical Python)**

NumPy (Numerical Python) - це потужна бібліотека для мови програмування Python, що надає ефективні структури даних у вигляді масивів фіксованого розміру. Вона дозволяє проводити швидкі обчислення для великої кількості числових даних і є ключовою бібліотекою для наукових обчислень у Python. Основним типом даних у NumPy є масив (ndarray), який представляє собою колекцію елементів одного типу даних у N вимірах. Масиви у NumPy є ефективними з точки зору використання пам'яті та швидкості, оскільки дозволяють проводити операції обробки даних без необхідності використання циклів. Бібліотека надає різноманіття вбудованих функцій для роботи з масивами, таких як математичні операції, операції з логічними значеннями, операції лінійної алгебри, індексація та зрізи масивів, а також операції зчитування та запису даних у файли. Крім того, NumPy інтегрується з іншими популярними бібліотеками для наукових обчислень, такими як SciPy, pandas і matplotlib.

### **Sklearn**

Sklearn, є однією з найпопулярніших та впливовіших бібліотек для машинного навчання у мові програмування Python. Ця бібліотека містить у собі різноманітні алгоритми машинного навчання, методи передобробки даних, валідації та оцінки моделей, а також інструменти для роботи з наборами даних. Sklearn побудована на основі інших ефективних бібліотек Python, таких як NumPy та SciPy, і має зручний інтерфейс для використання широкого спектру алгоритмів машинного навчання.

Sklearn реалізує алгоритми машинного навчання задля класифікації, регресії, кластеризації, зменшення розмірності та виявлення аномалій. Усе це містить у собі наступні методи, а саме:

1. метод опорних векторів;
2. дерева рішень;
3. випадкові ліси;

4. градієнтний бустінг;
5. наївний Баєс;
6. k-найближчі сусіди.

Крім того, бібліотека пропонує інструменти для передобробки даних, валідації моделей, пошуку оптимальних гіперпараметрів та оцінки якості моделей. Вбудований функціонал для роботи з текстовими даними, вибірками даних, перехресними перевірками та передобробкою даних, масштабуванням, кодуванням категоріальних ознак, заповненням пропущених значень та видаленням шуму. Бібліотека підтримує різні метрики для оцінки моделей, такі як точність, середньоквадратична помилка та інші.

У висновку можна зазначити потужність цієї бібліотеки задля виконання різноманітних завдань, оскільки велика кількість інструментів робить її універсальною для великої кількості завдань. Що стосується машинного навчання, то вона значно оптимізує та прискорює застосування алгоритмів.

### **OpenCV (OpenSource Computer Vision)**

OpenCV – це велика відкрита бібліотека комп'ютерного зору, яка надає велику кількість інструментів та засобів обробки зображень та відеоматеріалів. Бібліотек містить у собі наступні алгоритми комп'ютерного зору:

1. завантаження, обробка та збереження зображень чи відеоматеріалів;
2. обробка зображень, розмиття, фільтрація, видалення шуму;
3. розпізнавання об'єктів, символів чи певних шаблонів;
4. калібрування камери під час обробки відеоматеріалів;
5. перетворення кольорової гамми та її розпізнавання;
6. орієнтація у 3D просторі;

Це перелік самих базових алгоритмів бібліотеки. Слід зазначити, що OpenCV є доволі потужним інструментом під час обробки різноманітних даних, оскільки включає в себе велику кількість методів, алгоритмів які позитивно сприяють кінцевому результату. Оскільки бібліотека має відкритий код, то можна зазначити її доступність для всіх.

## **Matplotlib**

Matplotlib - це бібліотека для створення графіків та візуалізації даних у мові програмування Python. Вона надає широкий спектр можливостей для створення різноманітних типів графіків, включаючи лінійні графіки, стовпчасті графіки, точкові графіки, гистограми, кругові діаграми, контурні графіки та багато інших.

Matplotlib дозволяє контролювати всі аспекти графіків, від кольорів та стилів до осей та маркерів. Вона також підтримує використання різних типів даних, таких як списки, масиви NumPy, файли та дані, що генеруються динамічно.

Одним з ключових компонентів Matplotlib є модуль Pyplot, який забезпечує інтерфейс для створення графіків у стилі, схожому на MATLAB. Онлайн-документація Matplotlib та широкий спектр прикладів допомагають користувачам швидко вивчити та ефективно використовувати цю бібліотеку для візуалізації даних у Python. Matplotlib є однією з найпопулярніших бібліотек для візуалізації даних у Python і використовується в широкому спектрі галузей, включаючи науку, інженерію, фінанси, аналітику та багато інших.

### **3.2 Архітектура програмного забезпечення розпізнавання та сегментації об'єктів**

Розробка архітектури програмного забезпечення – це дуже важливий аспект будь-якого проекту. Архітектура програмного забезпечення визначає загальну структуру системи, включаючи розподіл функцій між її компонентами, способи комунікації між цими компонентами, та організацію даних.

Зазвичай архітектуру ПЗ відображають за допомогою різноманітних діаграм. На них зображують основні аспекти та базові методи відтворення застосунку. Враховуючи специфіку застосунку, що розробляється варто й тримати межу між функціоналом та зручністю виконання коду. Існує певна кількість вже базових принципів за якими й будується архітектура програмного забезпечення, а саме:

1. принцип інтерфейсів;
2. принцип пакування;
3. принцип розділення;

4. принцип відповідальності;
5. принцип Лісков.

Слідкування цим принципам гарантує правильну та зрозумілу розробку ПЗ.

Діаграма класів – це одна з головних UML компонентів, що допомагають у розробці будь-якого застосунку. Розробки діаграми дозволяє структурувати логічну базу даних, а також сформуванню загальний концепет застосунку. Надання статичного представлення застосунку є важливою частиною розробки. Загальний зовнішній вигляд діаграми дозволяє більш точно формувати уявлення про систему, адже усі елементи відображаються безпосередньо перед очима. Також прискорюється розробка та написання коду, що сприятливо впливає на загальну продуктивність роботи. Під час розробки проекту було розроблено діаграму класів, що зображена на рисунку 3.2.

Далі, наведено діаграму класів програмного забезпечення розпізнавання та сегментації зображення.

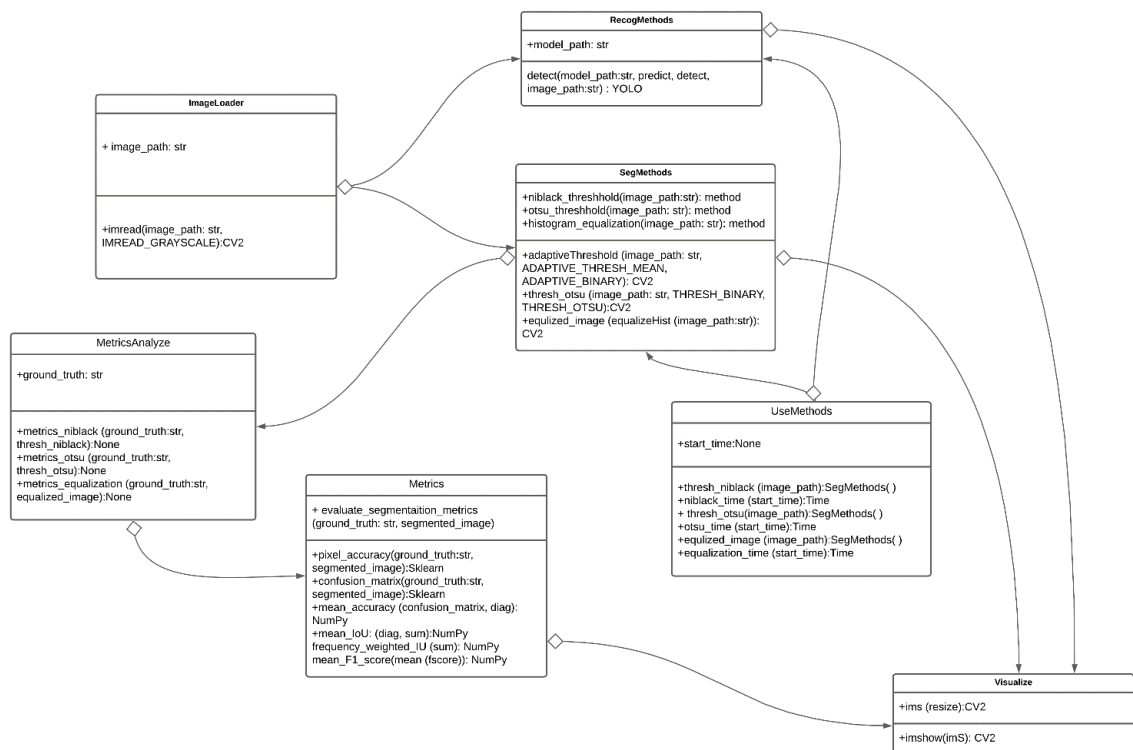


Рисунок 3.2 – Діаграма класів ПЗ

На діаграмі класів чітко відображенні усі компоненти застосунку, слід зазначити, що у подальшій розробці більшість з цих компонентів будуть змінені,

завдяки імплементації нових функцій та алгоритмів.

Також аналізуючи діаграму класів, можна відзначити, що під час розробки застосунку використовуються вбудовані бібліотеки Python, що свідчить можливість розширення функціоналу.

### **3.3 Аналіз та огляд наборів даних для розпізнавання та сегментації зображень**

Задля успішної роботи застосунків які повинні розпізнавати, сегментувати зображення існують спеціалізовані набори даних. Ці набори використовуються задля тренування моделей машинного навчання. Відомо, що для ефективною детекції об'єктів та елементів необхідна велика кількість навчальних даних (точок). Оскільки існують доволі специфічні та вузьконаправлені галузі, то виникає необхідність розмітки навчальної моделі власноруч. Створення анотацій для зображення містить у собі додавання певних метаданих, за допомогою яких об'єкти можуть бути ідентифіковані на зображенні. Вони можуть бути визначенні за допомогою особливостей об'єкту чи інших властивостей.

Наразі є доволі велика кількість інструментів для маркування та додавання метаданих на зображеннях. До цих інструментів відносять таких лідерів як: CVAT, LabelBox, SuperAnnotate, LabelStudio, тощо.

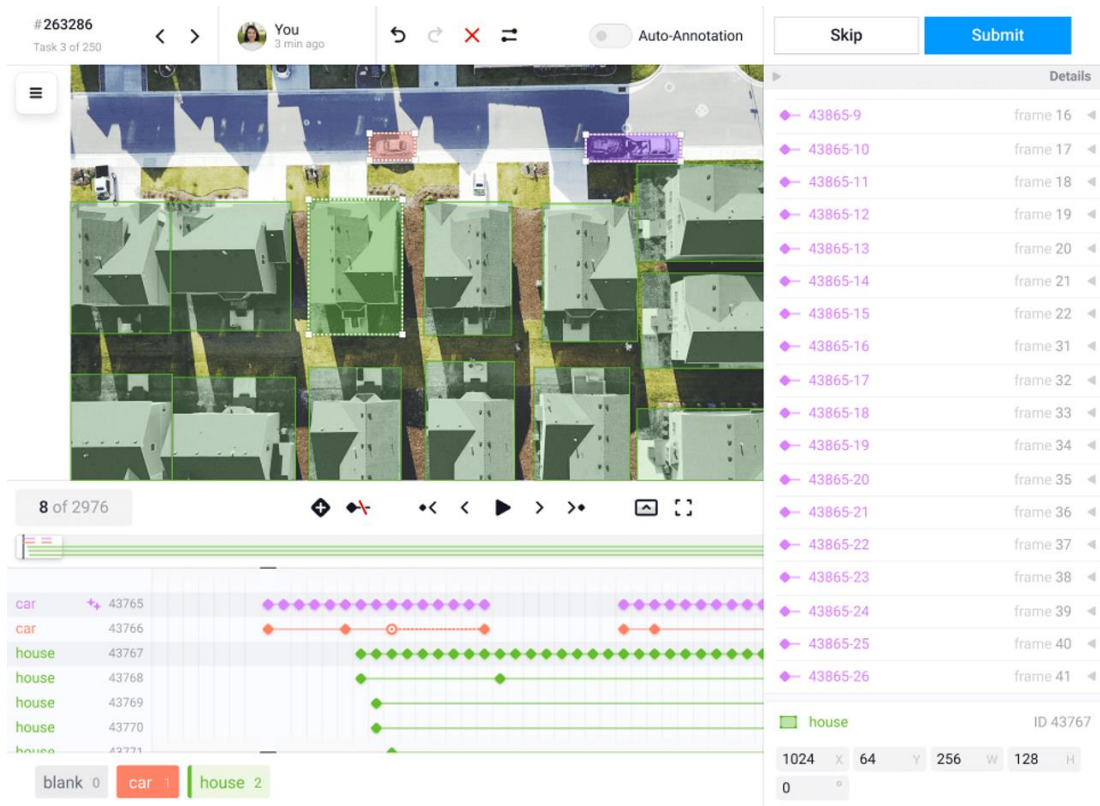


Рисунок 3.3 – Застосунок LabelStudio

Також є доволі велика кількість відкритих репозиторіїв з повністю розміченими зображеннями. Найпопулярнішою платформою є Kaggle. Він містить у своєму арсеналі тисячі готових навчальних даних. За допомогою цього ресурсу можна навчити модель розпізнавати як й медичні випадки, наприклад пухлини, тощо, так й загальні базові предмети побуту. Ще одним плюсом використання цієї платформи є те, що вона містить величезні розмічені набори даних, які були профінансовані великими компаніями з дата майнінгу, машиному навчнню, тощо. Такі дані використовуються для навчання найсучасніших моделей машинного навчання. Розглянемо декілька найпопулярніших наборів.

## COCO

Цей набір даних містить біля 300000 зображень, які мають велику кількість міток, що впливає на якість навчання моделі. Набір різноманітних анотацій допомагає використовувати його для найрізноманітніших завдань сегментації та розпізнавання зображень.



Рисунок 3.4 – COCO

До головних характеристик відносять:

1. різноманітність об'єктів: Набір даних COCO містить різноманітні типи об'єктів, включаючи людей, тварин, автомобілі, рослини та інші предмети, що зустрічаються в повсякденному житті;
2. різноманітність контекстів: Зображення в наборі даних COCO охоплюють різноманітні сцени та контексти, що варіюються від урбаністичних ландшафтів до природних пейзажів, внутрішніх приміщень та багато інших;
3. точність анотацій: Об'єкти на зображеннях в COCO анотовані з високою точністю та деталізацією, що дозволяє навчати моделі з високою точністю та ефективністю;
4. Масштабність: Набір даних COCO містить мільярди зображень з мільйонами анотацій, що робить його одним з найбільших та найпопулярніших джерел для завдань розпізнавання об'єктів та сегментації зображень;
5. широке використання: COCO використовується в академічних дослідженнях, промислових проектах та конкурсах, таких як COCO Captioning Challenge та COCO Detection Challenge.

Набір даних COCO є одним із стандартів у галузі комп'ютерного зору та обробки зображень, і використовується для навчання та оцінки різноманітних моделей машинного навчання, включаючи сверточні нейронні мережі (CNN) для розпізнавання об'єктів та сегментації зображень.

## **PASCAL VOC**

PASCAL VOC - це набір даних, який використовується для завдань розпізнавання об'єктів та сегментації зображень. Розроблений університетом Оксфорду, PASCAL VOC включає зображення об'єктів, що належать до 20 різних класів, таких як люди, тварини, транспортні засоби, меблі та інші.

Основні характеристики PASCAL VOC:

1. класи об'єктів: Набір даних PASCAL VOC містить зображення об'єктів, що належать до 20 різних класів, таких як люди, автомобілі, собаки, коти та інші;
2. масштабність: PASCAL VOC включає тисячі зображень з анотаціями об'єктів, що робить його досить розгалуженим та репрезентативним набором даних;
3. точність анотацій: Об'єкти на зображеннях в PASCAL VOC анотовані з високою точністю та деталізацією, що дозволяє ефективно навчати моделі та оцінювати їхню продуктивність;
4. використання в конкурсах: Набір даних PASCAL VOC використовується для проведення різних конкурсів та випробувань у галузі комп'ютерного зору та обробки зображень;
5. стандартні метрики оцінки: Для оцінки ефективності моделей, навчених на наборі даних PASCAL VOC, використовуються стандартні метрики, такі як точність розпізнавання об'єктів та середній показник точності;

PASCAL VOC є одним із широко використовуваних наборів даних у галузі комп'ютерного зору та розпізнавання об'єктів. Він дозволяє дослідникам та практикам у галузі ефективно розвивати та оцінювати моделі машинного навчання для різних задач обробки зображень.

## **Open Images**

Open Images - це великий та розгалужений набір даних зображень, який розроблений та підтримується компанією Google. Цей набір даних містить мільйони зображень різних об'єктів та сцен, анотованих з точними мітками та мітками класів. Open Images є одним з найбільших та найбільш розгалужених наборів даних у галузі комп'ютерного зору.



### Основні характеристики Open Images:

1. масштабність: Open Images містить мільйони зображень різної природи та з різних джерел, охоплюючи широкий спектр об'єктів, сцен та контекстів.

2. анотації та мітки класів: Кожне зображення в Open Images анотоване з точними мітками та мітками класів, що дозволяє ефективно навчати моделі розпізнавання об'єктів та сегментації зображень.

3. розмаїття об'єктів та сцен: Набір даних охоплює різноманітні об'єкти та сцени, включаючи людей, тварин, різні об'єкти, вулиці, природні ландшафти та інші.

4. використання у дослідженнях та проектах: Open Images є популярним джерелом даних для багатьох дослідницьких проектів та застосувань у галузі комп'ютерного зору, таких як розпізнавання об'єктів, сегментація зображень, відстеження об'єктів тощо.

5. підтримка спільноти та відкритий доступ: Open Images надає відкритий доступ до своїх даних та постійно оновлюється та поповнюється за участі спільноти дослідників та розробників.

Усі ці особливості роблять Open Images важливим ресурсом для досліджень у галузі комп'ютерного зору, де великі та розгалужені набори даних є ключовими для ефективного навчання та оцінки моделей машинного навчання.

### **SIPAKMED**

SIPAKMED - це база даних, яка містить зображення з попередньою обробкою та анотаціями, призначені для дослідження раку та розробки алгоритмів машинного навчання для його розпізнавання та діагностики. Це найбільший набір даних для цього конкретного захворювання.

### Основні характеристики бази даних SIPAKMED:

1. масштабність: База даних містить велику кількість зображень, які охоплюють різні стадії розвитку раку, а також здорові та аномальні зразки.

2. анотації та мітки класів: Кожне зображення в SIPAKMED анотоване з високою точністю та деталізацією, що дозволяє ефективно навчати моделі розпізнавання та класифікації.

3. різноманітність зразків: В базі даних представлені зразки з різними характеристиками та у різних умовах зйомки, що робить її більш репрезентативною для реальних умов захворювання.

4. використання у дослідженнях та медичних дослідженнях: База даних SIPAKMED використовується для проведення різних досліджень у галузі медичної діагностики та аналізу зображень, а також для розробки та валідації алгоритмів машинного навчання для автоматичного розпізнавання цервікального раку.

5. відкритий доступ: База даних є відкритою для використання дослідниками та спеціалістами у галузі медичної діагностики, що сприяє швидшому прогресу в цій області.

Завдяки такій базі даних вчені та медичні фахівці мають можливість ефективно досліджувати та розробляти нові методи діагностики та лікування раку, що важливо для покращення медичної практики та підвищення якості життя пацієнтів.

### **3.3 Визначення алгоритму навчання моделі YOLO**

Задля вдалої роботи з розпізнавання об'єктів на зображеннях варто визначити певні етапи та найголовніші аспекти. Перш за все треба визначити модель нейронної мережі яка буде найбільш ефективна. На другому етапі визначити тематику нейронної мережі, вона може бути як вузьконаправлена так й загальна. Після визначення тематики, треба обрати необхідний датасет який буде навчати модель за допомогою оброблених зображень, які містять усі необхідні анотації та метадані.

Перші три головних етапи вже були виконані, а саме зроблено порівняльний аналіз моделей нейронних мереж в результаті який було визначено кращу з них, а саме YOLO, визначено тематику, а саме загальні та всім відомі об'єкти побуту, тощо, розглянуто датасети з обробленими зображеннями.

YOLO - це досить популярний та ефективний алгоритм для розпізнавання об'єктів у зображеннях та відео. Основна особливість YOLO полягає в тому, що вона пропонує вирішення задачі розпізнавання об'єктів як задачу регресії на графіці зображення, що дозволяє отримати прогнози для всіх об'єктів одночасно, а не для кожного області окремо.

Навчання моделі YOLO (You Only Look Once) - це процес, який включає в себе підготовку набору даних, конфігурацію моделі, навчання та оцінку результатів. Нижче я надаю детальний опис кожного з цих кроків:

#### 1. підготовка набору даних:

- Збір та анотація зображень: Набір даних повинен містити зображення, на яких присутні об'єкти, що потрібно розпізнавати, а також анотації для цих об'єктів (bounding boxes та класи об'єктів).

- Перевірка та очищення даних: До навчання моделі важливо перевірити та очистити дані від помилок, дублікатів та інших аномалій.

#### 2. конфігурація моделі YOLO:

- Вибір версії моделі: Існують різні версії YOLO, такі як YOLOv3, YOLOv4 тощо. Вибір версії залежить від ваших потреб, обчислювальних можливостей та інших факторів.

- Вибір параметрів навчання: Це включає в себе вибір швидкості навчання (learning rate), розміру міні-пакетів (batch size), кількості епох (epochs) та інших гіперпараметрів.

#### 3. навчання моделі:

- Завантаження переднього зображення (pre-trained weights): Для прискорення процесу навчання можна використовувати передні ваги (pre-trained weights), які вже навчалися на великому наборі даних.

- Завантаження та налаштування архітектури мережі: Використовуючи фреймворк для глибокого навчання (наприклад, TensorFlow або PyTorch), завантажте архітектуру моделі YOLO та налаштуйте її для вашого набору даних.

- Навчання моделі: Запустіть процес навчання, передавши зображення та відповідні анотації моделі YOLO та адаптуючи її ваги для вашого конкретного завдання.

#### 4. оцінка та тестування моделі:

- Оцінка продуктивності: Після завершення навчання моделі оцініть її продуктивність, використовуючи метрики, такі як точність розпізнавання та середня точність на тестовому наборі даних.

- Тестування на нових зображеннях: Перевірте роботу навченої моделі на нових зображеннях, щоб переконатися, що вона ефективно розпізнає об'єкти на реальних даних.

5. тонке налаштування та оптимізація:

- Якщо результати навчання не задовольняють ваших очікувань, ви можете внести корективи до параметрів навчання, архітектури моделі та інших налаштувань, щоб покращити її продуктивність.

### 3.4 Метрики сегментації

Метрика - це числова міра, яка використовується для вимірювання або оцінки явища, процесу або характеристики. У контексті обробки зображень та комп'ютерного зору метрики можуть використовуватися для оцінки якості роботи алгоритмів сегментації, класифікації, виявлення об'єктів та інших завдань.

#### Метрика Pixel accuracy

Метрика Pixel accuracy - це числова міра, яка використовується для оцінки якості сегментації зображень у задачах комп'ютерного зору. Ця метрика вимірює відсоток пікселів, які були правильно класифіковані алгоритмом сегментації.

Простий спосіб обчислення точності пікселів полягає у поділі кількості правильно класифікованих пікселів на загальну кількість пікселів у зображенні. Це дає нам відсоток правильно класифікованих пікселів.

Математично формулу для обчислення точності пікселів можна записати так:

$$\text{Точність пікселів} = \frac{\text{Кількість правильно класифікованих пікселів}}{\text{Загальна кількість пікселів у зображенні}} * 100 \quad (4.1)$$

Ця метрика корисна, коли кожен піксель на зображенні має значення, і точність класифікації на рівні пікселів є ключовим аспектом оцінки якості роботи алгоритму. Однак її недолік полягає в тому, що вона не враховує просторову структуру об'єктів на зображенні і може дати неправильні оцінки, якщо об'єкти на зображенні несбалансовані за розміром або розподілом по зображенню.

## Метрика Mean accuracy

Метрика "Mean accuracy" (середня точність) - це метрика, яка використовується для оцінки якості класифікації в задачах машинного навчання, особливо в контексті класифікації зображень.

Зазвичай, середня точність обчислюється як середнє арифметичне точності для кожного класу. Для кожного класу точність визначається як відношення кількості правильно класифікованих екземплярів цього класу до загальної кількості екземплярів цього класу.

Середня точність дає загальну інформацію про ефективність класифікації по всім класам, незалежно від їх розміру або розподілу. Ця метрика корисна для порівняння різних моделей або підходів до класифікації, оскільки вона дає узагальнену оцінку їхньої ефективності.

## Метрика Mean F-1 score

Метрика "Mean F1-score" - це середнє арифметичне значення F1-показника для кожного класу в задачі багатокласової класифікації. F1-показник є комбінацією точності (precision) і повноти (recall) і використовується для оцінки якості класифікації моделі.

F1-показник обчислюється за такою формулою:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (4.2)$$

де:

- precision - це відношення правильно класифікованих екземплярів певного класу до усіх екземплярів, які були класифіковані як цей клас;
- recall - це відношення правильно класифікованих екземплярів певного класу до усіх екземплярів даного класу у вибірці.

Потім Mean F1-score обчислюється як середнє арифметичне значення F1-показника для всіх класів. Ця метрика дає загальне уявлення про ефективність класифікації моделі для всіх класів разом.

Mean F1-score є важливою метрикою у багатокласовій класифікації, оскільки вона враховує як точність, так і повноту, що дає більш повне уявлення про

продуктивність моделі.

### **Метрика Mean IoU**

Метрика Mean Intersection over Union - це метрика, що використовується для оцінки якості сегментації зображень у задачах комп'ютерного зору. Вона вимірює ступінь перекриття між сегментаційними масками для оцінки того, наскільки точно сегментація моделі визначає реальні області об'єктів на зображенні.

Формула для обчислення IoU для кожного окремого класу (наприклад, у багатокласовій сегментації) полягає у співвідношенні перетину та об'єднання прогнозованих та справжніх сегментаційних масок для кожного пікселя:

Mean IoU є важливою метрикою в сегментації зображень, оскільки вона враховує як перекриття, так і взаємозв'язок між прогнозованими та справжніми сегментаційними масками. Вона дозволяє отримати узагальнену інформацію про точність сегментації моделі для різних класів об'єктів на зображенні.

### **Метрика Frequency weighted IU**

Метрика Frequency weighted - це метрика, яка використовується для оцінки якості сегментації в задачах комп'ютерного зору. Вона є модифікацією середнього значення індексу перетину (Intersection over Union, IoU), що приділяє більшу вагу менш репрезентованим класам, забезпечуючи більш об'єктивну оцінку якості моделі.

Для розрахунку Frequency weighted IU спочатку обчислюється індекс перетину (IoU) для кожного окремого класу (наприклад, у багатокласовій сегментації) за допомогою наступної формули:

$$IoU = \frac{\text{Площа перетину масок}}{\text{Площа об'єднання масок}} \quad (4.3)$$

Потім вага IoU для кожного класу обчислюється, використовуючи його частоту в наборі даних, наприклад, шляхом визначення відношення кількості пікселів, які належать кожному класу, до загальної кількості пікселів у наборі даних.

Нарешті, Frequency weighted IU обчислюється як середньозважене значення IoU для всіх класів, де кожне значення IoU виважується його частотою у наборі даних.

Ця метрика дозволяє врахувати нерівномірність розподілу класів у наборі даних, забезпечуючи більш об'єктивну оцінку якості сегментації моделі..

### **Висновок до розділу 3**

Визначено та програмні методи для розробки застосунку розпізнавання та сегментації об'єктів на зображеннях. Розглянуто бібліотеки для обробки даних та основні методи, а саме: NumPy, Sklearn, OpenCV, Matplotlib.

Також розглянуто різноманітні моделі з навченими моделями задля розпізнавання та сегментації. Проведено аналіз методів сегментації на базі різноманітних даних та визначено їх метрики.

Аналіз методів сегментації на базі різноманітних даних допомагає визначити найефективніші підходи для конкретних завдань. Метрики, такі як точність пікселя, середня IoU та інші, можуть бути використані для об'єктивної оцінки результатів роботи алгоритмів сегментації.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ З СЕГМЕНТАЦІЇ ТА РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

### 4.1 Розробка структури застосунку

Проект розроблено у певному стилі, що відокремлює деякі функціональні модулі. Методології є важливим аспектом для успішної розробки, розбиття та структурування проекту робить його більш зрозумілим та легшим у сприйнятті. Нижче наведена структура проекту рис. 4.1.

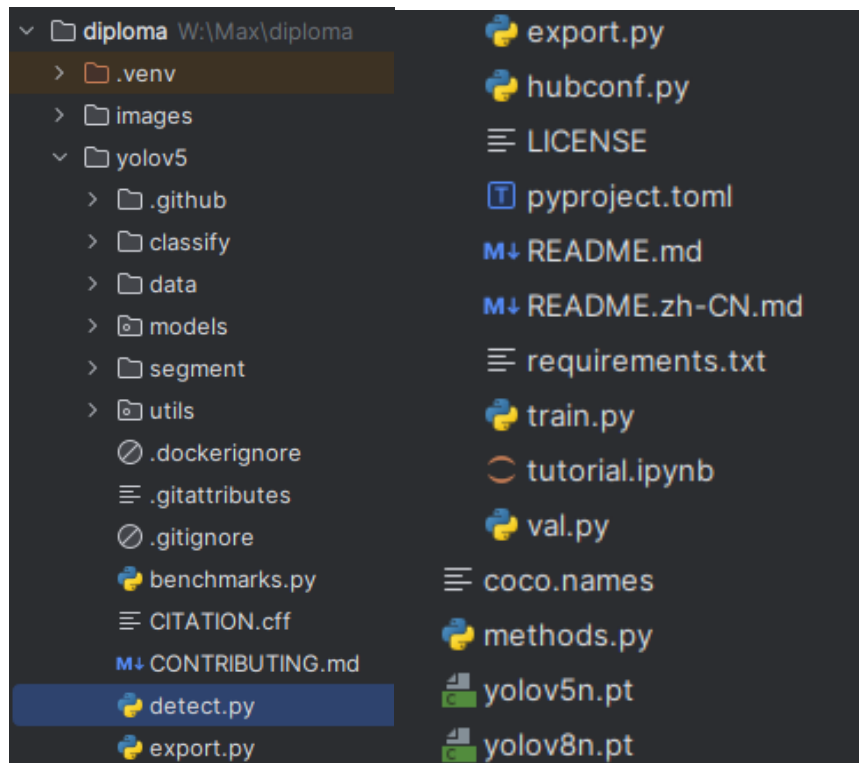


Рисунок 4.1 – Структура проекту

Проаналізуємо директорії та модулі, що використовуються у проекті.

- Директорія images містить у собі підготовлені моделі, які містять усі необхідні анотації для сегментації та розпізнавання зображень.

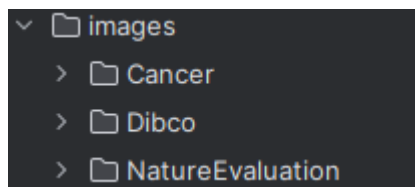


Рисунок 4.2 – Директорія images

Кожна з моделей розташована в окремих директоріях також, що у свою чергу надає спрощене розуміння структури.



– Директорія yolov5 – це директорія, що використовується безпосередньо для розпізнавання об'єктів на зображеннях. Директорія містить у собі усі необхідні алгоритми та методи, що надають можливість вдало використовувати їх задля позитивного результату. Слід зазначити, що ця директорія є у відкритому доступі як Git – репозиторій. Файл detect.py відповідає безпосередньо за обробку та детектування даних. Файл benchmarks.py використовується для збору метрик та статистичних даних, щодо швидкодії, точності, тощо.

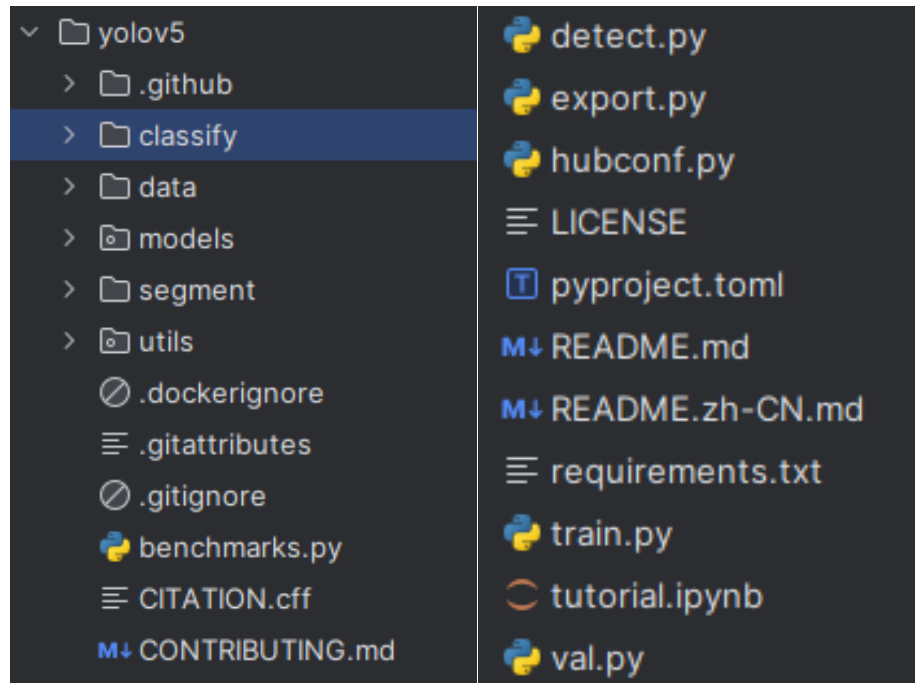


Рисунок 4.2 – Директорія yolov5

– Головний модуль methods.py, що містить у собі усі методи, алгоритми та функції застосунку. Він відповідає за навчання моделі, імпортування усіх необхідних бібліотек, впровадження алгоритмів сегментації та розпізнавання, а також розрахунку метрик використання.

– Модель yolovn5.pt – це файл який містить у собі параметри задля навчання моделі, що були навчені заздалегідь на певному наборі даних для розпізнавання об'єктів.

## 4.2 Кодування програмного застосунку

Кодування – це одна з важливих частин розробки програмного забезпечення, оскільки вона містить у собі реалізацію всієї логіки та візуально відображає

реалізовані алгоритми. Під час розробки було визначено основні етапи та частини коду. Насперед раніше розроблені діаграми використання значно спроситили цей етап.

Отже розглянемо основну структуру коду застосунку. Вона розбита на певні модулі, який відповідає за власну функцію. Також є певний блок, що імпортує усі необхідні бібліотеки та інструменти.

```
import cv2
import numpy as np
import time
import torch
from sklearn.metrics import accuracy_score, confusion_matrix, precision_recall_fscore_support
```

Рисунок 4.3 – Імпортування бібліотек

Функція, що підвантажує зображення задля розпізнавання та сегментації об'єктів.

```
# Зчитування зображення
image = cv2.imread(filename='images/images.jpg', cv2.IMREAD_GRAYSCALE)
```

Рисунок 4.4 – Функція зчитування зображення

Можна зазначити, що зчитування зображення здійснюється за допомогою вбудованих функцій у бібліотеку OpenCV.

Після зчитування зображення було проініціалізовано алгоритми сегментації зображення. Усі вони винесені в окремі функції, що мають назви відповідні до самих алгоритмів та методів, а саме `niblack_threshold`, `otsu_threshold`, `histogram_equalization`.

```
def niblack_threshold(image):
    thresh_niblack = cv2.adaptiveThreshold(image, maxValue=255, cv2.ADAPTIVE_PRETHRESH_MEAN_C, cv2.THRESH_BINARY, blockSize=11, -10)
    return thresh_niblack
# Порогова бінаризація за допомогою методу Оцу
1 usage
def otsu_threshold(image):
    _, thresh_otsu = cv2.threshold(image, thresh=0, maxval=255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    return thresh_otsu
# Балансування гистограми
1 usage
def histogram_equalization(image):
    equalized_image = cv2.equalizeHist(image)
    return equalized_image
```

Рисунок 4.5 – Методи сегментації

Кожен з цих методів отримує раніше ініціалізоване зображення, оброблює завдяки вбудованим методам у бібліотеку та повертає у спеціальний масив вже оброблене зображення. Слід зазначити, що для методів Ніблека та Отсу варто

використовувати функції, що обробляють зображення більш детально.

Наступним етапом розробки програмного забезпечення є визначення метрик сегментації, вони також виокремлені в окрему функцію, що виглядає наступним чином.

```
def evaluate_segmentation_metrics(ground_truth, segmented_image):
    pixel_accuracy = accuracy_score(ground_truth.flatten(), segmented_image.flatten())
    confusion_mat = confusion_matrix(ground_truth.flatten(), segmented_image.flatten())
    true_positive = np.diag(confusion_mat)
    false_positive = np.sum(confusion_mat, axis=0) - true_positive
    false_negative = np.sum(confusion_mat, axis=1) - true_positive
    true_negative = np.sum(confusion_mat) - (true_positive + false_positive + false_negative)
    mean_accuracy = np.mean(true_positive / (true_positive + false_positive + false_negative))
    mean_IoU = np.mean(true_positive / (true_positive + false_positive + false_negative))
    frequency_weighted_IU = np.sum(true_positive) / (
        np.sum(true_positive + false_positive) + np.sum(true_positive + false_negative))
    precision, recall, fscore, _ = precision_recall_fscore_support(ground_truth.flatten(), segmented_image.flatten(),
                                                                    average='weighted')
    mean_F1_score = np.mean(fscore)

    return pixel_accuracy, mean_accuracy, mean_IoU, frequency_weighted_IU, mean_F1_score
```

Рисунок 4.6 – Функція визначення метрик

Наведена функція містить у собі змінні, що визначають точність пікселі, точність сегментації за використаними класами, оцінки наскільки добре модель виділяє об'єкти, а також виділення об'єктів з незбалансованими класами, показник між точністю та повнотою сегментації. Кожна з цих метрик оцінює різні аспекти якості сегментації моделі і їх поєднання може дати більш повне уявлення про її продуктивність. Ця функція повертає результати метрик.

Далі після визначення метрик визначається час сегментації за кожним методом, задля визначення швидкодії того чи іншого методу. Ця функція використовує бібліотеку `time`, що використовує системний годинник.

```
start_time = time.time()
thresh_niblack = niblack_threshold(image)
niblack_time = time.time() - start_time

start_time = time.time()
thresh_otsu = otsu_threshold(image)
otsu_time = time.time() - start_time

start_time = time.time()

unet_time = time.time() - start_time

start_time = time.time()
equalized_image = histogram_equalization(image)
equalization_time = time.time() - start_time
```

Рисунок 4.7 – Функція визначення швидкодії

Наступна функція оцінює метрики для кожного з методів. Оцінка базується на порівнянні вже підготовленої моделі із зчитаним зображенням.

```
ground_truth = cv2.imread( filename: 'images/images.jpg', cv2.IMREAD_GRAYSCALE)
metrics_niblack = evaluate_segmentation_metrics(ground_truth, thresh_niblack)
metrics_otsu = evaluate_segmentation_metrics(ground_truth, thresh_otsu)

metrics_equalization = evaluate_segmentation_metrics(ground_truth, equalized_image)
```

Рисунок 4.8 – Функція оцінки метрик

Функція візуалізації результатів сегментації виглядає наступним чином.

```
Hori = np.concatenate( arrays: (image, thresh_niblack, thresh_otsu, equalized_image), axis=1)
imS = cv2.resize(Hori, dsize: (1768, 640))
cv2.imshow( winname: 'Original, Niblack, Otsu, Histogram Equalization segmentation methods', imS)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Рисунок 4.9 – Функція візуалізації сегментації

Функція використовує певні методи, які розташовують вже сегментовані зображення у більш гарному положенні, також є метод який редагує розмір оброблених зображень.

Функція розпізнавання зображення містить у собі метод ініціалізації зображення, а також встановлення необхідного розміру. Функція використовує методи бібліотеки OpenCV, що дозволяють виділяти об'єкти. Змінна model визначає навчену модель, що містить у собі усі необхідні параметри та налаштування.

```
model = torch.hub.load(repo_or_dir='yolov5', model='yolov5n', source='local')

img = cv2.imread('images/images.jpg')
img = cv2.resize(img, dsize=(1000,650))
result = model(img)
print(result)
df = result.pandas().xyxy[0]
print(df)

for ind in df.index:
    x1, y1 = int(df['xmin'][ind]), int(df['ymin'][ind])
    x2, y2 = int(df['xmax'][ind]), int(df['ymax'][ind])
    label = df['name'][ind]
    cv2.rectangle(img,(x1,y1), (x2,y2), (255,255,0),2)
    cv2.putText(img,label,org=(x1,y1-5), cv2.FONT_HERSHEY_PLAIN, fontScale=2, color=(255,255,0), thickness=2)

cv2.imshow(winname='IMAGE',img)
cv2.waitKey(0)
```

Рисунок 4.10 – Функція візуалізації сегментації

Кожна метрика виводиться окремо для кожного методу у вікно консолі.

```
print('Metrics for Niblack Thresholding:')
print('Pixel accuracy:', metrics_niblack[0])
print('Mean accuracy:', metrics_niblack[1])
print('Mean IoU:', metrics_niblack[2])
print('Frequency weighted IU:', metrics_niblack[3])
print('Mean F1-score:', metrics_niblack[4])
print('')
print('Metrics for Otsu Thresholding:')
print('Pixel accuracy:', metrics_otsu[0])
print('Mean accuracy:', metrics_otsu[1])
print('Mean IoU:', metrics_otsu[2])
print('Frequency weighted IU:', metrics_otsu[3])
print('Mean F1-score:', metrics_otsu[4])
print('')
print('Metrics for Histogram Equalization:')
print('Pixel accuracy:', metrics_equalization[0])
print('Mean accuracy:', metrics_equalization[1])
print('Mean IoU:', metrics_equalization[2])
print('Frequency weighted IU:', metrics_equalization[3])
print('Mean F1-score:', metrics_equalization[4])
```

Рисунок 4.11 – Функція візуалізації сегментації

Значення метрик використовуються з раніше створених масивів метрик для кожного з методів.



### 4.3 Тестування програмного забезпечення

Під час використання перш за все варто, завантажити зображення чи певний набір для обробки. Тестування застосунку проводилося на 3 датасетах, що дозволять визначити метрики для методів сегментації.

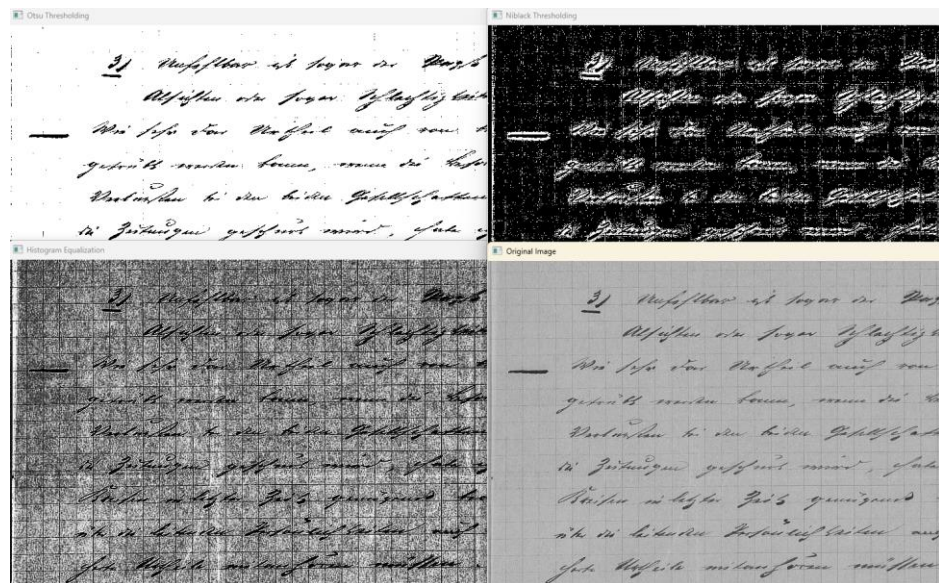


Рисунок 4.12 – Результати сегментації DIBCO2019

```
Metrics for Niblack Thresholding:  
Pixel accuracy: 0.0022016651248843665  
Mean accuracy: 5.686579047879866e-05  
Mean IoU: 5.686579047879866e-05  
Frequency weighted IU: 0.0011008325624421833  
Mean F1-score: 0.0001762679370764128  
  
Metrics for Otsu Thresholding:  
Pixel accuracy: 0.006337566226557901  
Mean accuracy: 2.6941781726951046e-05  
Mean IoU: 2.6941781726951046e-05  
Frequency weighted IU: 0.0031687831132789506  
Mean F1-score: 8.412757805657193e-05  
  
Metrics for Histogram Equalization:  
Pixel accuracy: 0.006610041207636027  
Mean accuracy: 0.004204465127238454  
Mean IoU: 0.004204465127238454  
Frequency weighted IU: 0.0033050206038180135  
Mean F1-score: 0.00637621854436408
```

Рисунок 4.13 – Метрики сегментації датасету DIBCO2019

Можна зазначити, що метод Отсу є найкращим рішенням для сегментації рукописних документів.

Далі було протестовано датасет, який містить у собі певні шуми та зернистість, що в свою чергу ускладнює розпізнавання.

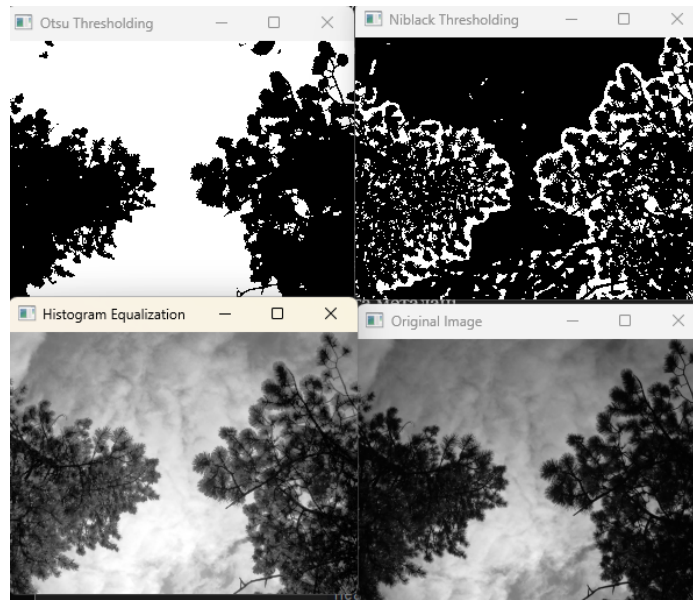


Рисунок 4.14 – Результати сегментації NatureEvaluaton

```
Metrics for Niblack Thresholding:
Pixel accuracy: 0.0
Mean accuracy: 0.0
Mean IoU: 0.0
Frequency weighted IU: 0.0
Mean F1-score: 0.0

Metrics for Otsu Thresholding:
Pixel accuracy: 0.0
Mean accuracy: 0.0
Mean IoU: 0.0
Frequency weighted IU: 0.0
Mean F1-score: 0.0

Metrics for Histogram Equalization:
Pixel accuracy: 0.003096296296296296
Mean accuracy: 4.7241727329339264e-05
Mean IoU: 4.7241727329339264e-05
Frequency weighted IU: 0.001548148148148148
Mean F1-score: 0.006067465637483919
```

Рисунок 4.15 – Метрики сегментації датасету NatureEvaluaton

Можна зазначити, що два методи не змогли доцільно визначити об'єкти, можливо це через доволі велику зернистість зображення.

Наступний протестований датасет містить медичні знімки на який зображені пухлини головного мозку. Цей тест є дуже важливим оскільки виявлення та виділення таких об'єктів є дуже необхідним у сучасній медицині.

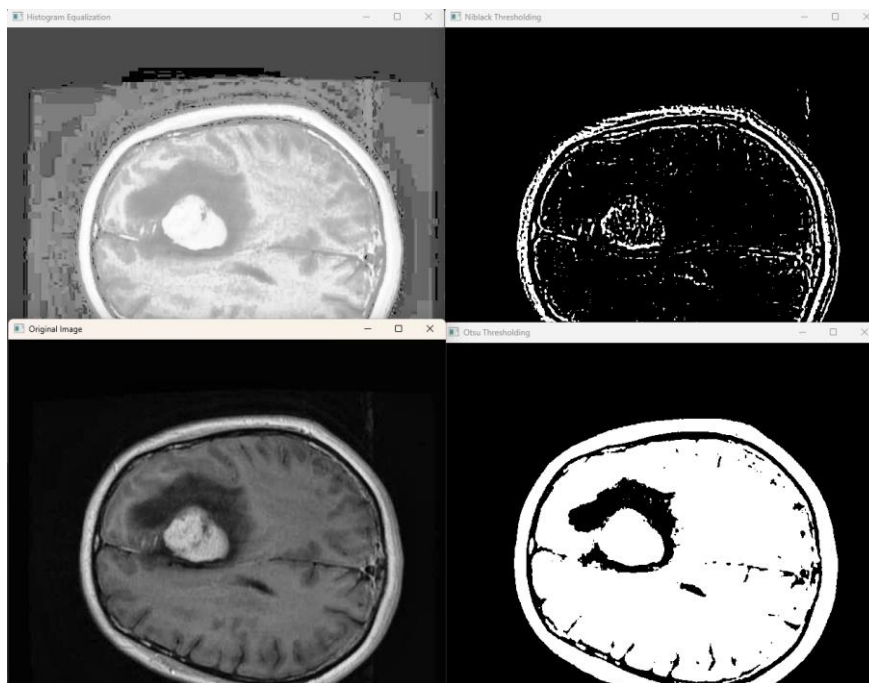


Рисунок 4.16 – Результати сегментації SirakMed

```
Metrics for Niblack Thresholding:  
Pixel accuracy: 0.00024658203125  
Mean accuracy: 1.0397148422484934e-06  
Mean IoU: 1.0397148422484934e-06  
Frequency weighted IU: 0.000123291015625  
Mean F1-score: 1.4013201783163403e-07  
  
Metrics for Otsu Thresholding:  
Pixel accuracy: 0.000234375  
Mean accuracy: 1.4110708620367486e-06  
Mean IoU: 1.4110708620367486e-06  
Frequency weighted IU: 0.0001171875  
Mean F1-score: 1.8624576688114334e-07  
  
Metrics for Histogram Equalization:  
Pixel accuracy: 0.00280517578125  
Mean accuracy: 0.001511408667357149  
Mean IoU: 0.001511408667357149  
Frequency weighted IU: 0.001402587890625  
Mean F1-score: 0.002370669498297588
```

Рисунок 4.17 – Метрики сегментації датасету SirakMed

Вище наведені результати сегментації. Можна зазначити, що метод гістограм є найкращим.



Розглядаючи тестування функції розпізнавання було відібрано 3 зображення різноманітної складності, а саме з маленькою, середньою, великою кількістю об'єктів на зображеннях.

Перше тестування - це тестування зображення з маленькою та чіткою кількістю об'єктів на зображенні.

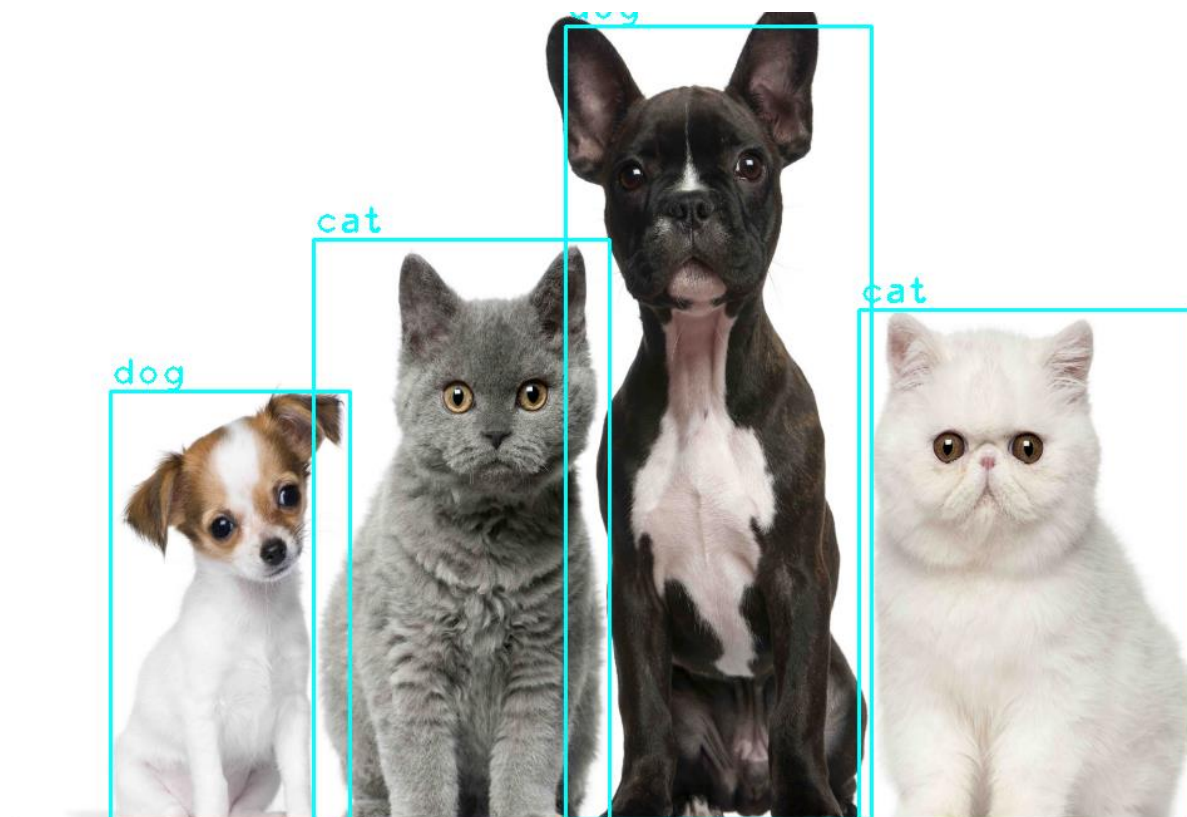


Рисунок 4.18 – Результат розпізнавання

Можна зазначити, що застосунок впорався з цим завданням на всі 100% та безпомилково визначив об'єкти.

```
image 1/1: 650x1000 2 cats, 2 dogs
Speed: 4.0ms pre-process, 99.1ms inference, 2.0ms NMS per image at shape (1, 3, 416, 640)
```

	xmin	ymin	xmax	ymax	confidence	class name
0	81.994568	304.886810	272.417236	650.000000	0.593274	16 dog
1	678.857605	239.379196	946.886169	647.810364	0.437826	15 cat
2	243.346695	183.997635	479.878540	647.339661	0.365793	15 cat
3	444.258118	13.334465	688.053223	645.806641	0.296601	16 dog

Рисунок 4.19 – Метрики розпізнавання

Розглядаючи метрики, можна зазначити чітке виділення 4 об'єктів та їх характеристик.

Друге тестування – це тестування зображення яке містить середню кількість об'єктів.



Рисунок 4.20 – Результат розпізнавання

Друге тестування визначило більшість об'єктів з точністю 80%, але все ж таки цей результат варто вважати прийнятним.

```
image 1/1: 650x1000 2 cups, 1 chair, 1 potted plant, 2 tvs, 1 mouse, 1 keyboard, 1 cell phone
Speed: 4.0ms pre-process, 93.5ms inference, 1.0ms NMS per image at shape (1, 3, 416, 640)
  xmin      ymin      xmax      ymax  confidence  class \
0  53.054249  253.618729  269.091766  391.627747  0.780324  62
1  0.000000  349.022888  242.449860  647.532410  0.678949  56
2  419.565247  363.724060  602.998108  427.661438  0.603709  66
3  318.293243  363.716522  357.730927  407.687866  0.536552  41
4  372.210022  190.930756  677.543945  333.686371  0.506817  62
5  721.961487  334.164520  790.337463  409.962097  0.360318  58
6  721.912598  338.209717  790.265503  409.763550  0.337070  41
7  602.315796  357.458130  686.394104  389.728210  0.309125  67
8  640.046936  424.011658  706.686462  451.147430  0.294604  64

      name
0      tv
1     chair
2   keyboard
3       cup
4        tv
5  potted plant
6       cup
7   cell phone
8       mouse
```

Рисунок 4.21 – Метрики розпізнавання

Розглядаючи метрики, можна зазначити чітке виділення 8 об'єктів та їх характеристик, але все ж з певною помилкою, адже позиція 7 є лампою.

Третє тестування – це тестування зображення яке містить велику кількість об'єктів.



Рисунок 4.22 – Результат розпізнавання

Третє тестування визначило зовсім малу кількість об'єктів. Причиною цього є велика кількість шумів, підвищення зернистість, а також зливання кольорових палітр.

```
image 1/1: 650x1000 2 persons, 2 cars, 1 motorcycle
Speed: 5.0ms pre-process, 85.5ms inference, 1.0ms NMS per image at shape (1, 3, 416, 640)
      xmin      ymin      xmax      ymax  confidence  class \
0  464.071472  610.008545  550.950134  649.937927   0.469683   2
1  759.896729  589.207642  783.386475  650.000000   0.357845   0
2  742.143860  589.458130  765.585693  648.821045   0.278801   0
3  330.331543  607.913513  427.456909  649.885010   0.271186   3
4  254.223419  589.509521  344.401215  647.504517   0.266377   2

      name
0      car
1     person
2     person
3 motorcycle
4      car
```

Рисунок 4.23 – Метрики розпізнавання

Розглядаючи метрики, можна зазначити, виділення об'єкти розпізнані чітко, але аналізуючи зображення можна зробити висновок про низьку точність розпізнавання об'єктів в цілому.



## **Висновки до розділу 4**

У ході виконання розділу було розроблено програмне забезпечення розпізнавання та сегментації зображень з допомогою комп'ютерного зору.

Також було розроблено та оглянуто структуру програмного забезпечення, що в свою чергу дає низку переваг під час основної програмної розробки.

Було проведено огляд програмного коду з детальним поясненням кожної функції та алгоритмів роботи того чи іншого методу. Такий детальний аналіз відображає усю сутність розробки програмного забезпечення за цією тематикою.

Наприкінці було протестовано розроблене програмне забезпечення та наведено відповідні скріншоти. Також було проаналізовано отримані метрики як для сегментації так й для розпізнавання. Тестування сегментації проводилося за рахунок 3 різних датасетів, що містять різні за типом та складом зображення. Таке тестування дозволяє визначити найоптимальніший метод для цієї чи іншої ситуації.

Насамперед тестування розпізнавання також проводилося на базі 3 зображень, що містять різну кількість об'єктів.

## ВИСНОВКИ

За результатом виконання кваліфікаційної роботи магістра було розроблено програмне забезпечення розпізнавання та сегментації об'єктів на зображеннях. Задля цього та для розуміння вимог та загальних тенденцій даної сфери, було проаналізовано та описано предметну сферу. Було проведено аналіз та дослідження нейронних мереж, методів розпізнавання та сегментації. Також було визначено проблематику обробки зображень. Визначення проблематики допомогло оцінити методи та обрати найкращі з них. Було розглянуто існуючі рішення виконання цієї задачі, а саме нейронні мережі, шаблонний метод, Віола Джонса, тощо. Після детального аналізу було визначено специфікацію вимог застосунку.

Визначені математичні моделі дозволили поглибитись у сутність алгоритмів нейронних мереж та більш детально розглянути кожен з них. Задля виокремлення найкращого методу розпізнавання зображень, було проведено невелике тестування на швидкодію та точність. З допомогою цього визначено найкращу архітектуру YOLO. Після усіх підготовчих операцій було розроблено діаграму використання на якій зображені варіанти використання застосунку та його можливості.

Застосунок розроблено на мові програмування Python, використано бібліотеки OpenCV, NumPy, Time, тощо. Архітектура програмного забезпечення відтворена та спроектована на діаграмі класів, що містить у собі усі необхідні дані для розуміння взаємозв'язків системи. Окремо слід виділити процес аналізу наборів даних для розпізнавання та сегментації. Набори COCO, PASCAL VOC, Open Images, SIPAKMED є базовими у цій розробці.

Набутий досвід позитивно сприятиме для подальших розробок та поліпшення вже наявних проектів включаючи й даний проект. Слід зазначити, що деякі функції можуть бути імплементовані в цей проект у майбутньому.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про комп'ютерний зір. URL <https://www.simplilearn.com/computer-vision-article> (дата звернення: 17.02.2024)
2. M. Rever. Computer Vision Projects with OpenCV and Python 3. 20. P. 62-63. (дата звернення: 17.02.2024)
3. What is computer vision? URL: <https://www.ibm.com/topics/computer-vision> (дата звернення: 17.02.2024)
4. M. Kumar, R. K. Sharma, I. Sethi. Machine Learning in Image Analysis and Pattern Recognition. 2021. P. 44-52.
5. Про розпізнавання зображень.  
URL: <https://www.techtarget.com/searchenterpriseai/definition/image-recognition> (дата звернення: 17.02.2024)
6. Про глибинне машинне навчання URL: <https://www.sciencedirect.com/topics/engineering/deep-learning> (дата звернення: 17.02.2024)
7. T. Blashke, C. Burnett Image Segmentation Methods for Object-based Analysis and Classification. 2020. P. 211-212. (дата звернення: 17.02.2024)
8. G. Carneiro, L. Yang. Y. Zheng. Deep Learning and Convolutional Neural Networks for Medical Image Computing. 2017. P. 30-32.
9. Y. Goldberg. Neural Network Methods in Natural Language Processing. 2017. P. 10-30 .
10. A. Bielecki. Models of Neurons and Perceptrons: Selected Problems and Challenges. 2018. C. 15-17.
11. Bento C. Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis . 2021.
12. Ferrari V., Tuytelaars T., Gool L. Van. Simultaneous object recognition and segmentation from single or multiple model views. International Journal of Computer Vision. 2006. P. 159–163.
13. Najman L., Challa A., Danda S., Sagar D. Power Spectral Clustering. Journal of

Mathematical Imaging and Vision. 2018.

14. D. Graupe. Principles Of Artificial Neural Networks: Basic Designs To Deep Learning (4th Edition). 2019. P. 20-30.

15. R. Clette. Комп'ютерне бачення. Теорія та алгоритми. користувача 2022.  
С. 43-44.

16. International Science Group. Multidisciplinary academic notes. Theory, methodology and practice. 2022. P. 68-69.

17. Про розпізнавання з допомогою моделі YOLO URL:  
<https://pjreddie.com/darknet/yolo/> (дата звернення: 17.02.2024)

18. Про розпізнавання з допомогою моделі HOG URL:  
<https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f> (дата звернення: 17.02.2024)

19. Про датасет DIBCO2019 URL: <https://paperswithcode.com/dataset/dibco-and-h-dibco> (дата звернення: 18.02.2024)

20. Набори датасетів URL: <https://www.kaggle.com/datasets> (дата звернення: 18.02.2024)

21. Про бібліотеку OpenCV URL: <https://opencv.org/> (дата звернення: 18.02.2024)

22. Про бібліотеку NumPy URL: <https://numpy.org/> (дата звернення: 18.02.2024)

23. Про діаграму використання. URL:  
<https://p4ilka.blogspot.com/2018/12/uml.html> (дата звернення: 18.02.2024)

24. Про датасет SIPAKMED URL: <https://ieeexplore.ieee.org/document/8451588>  
(дата звернення: 24.05.2022)

## ДОДАТОК А

### Матеріали апробації роботи

---

Міністерство освіти і науки України  
Чорноморський національний університет  
імені Петра Могили



### «Інформаційні технології та інженерія»

*Всеукраїнська науково-практична конференція  
молодих вчених, аспірантів і студентів*

## ТЕЗИ

*31 січня – 2 лютого 2024 року*

Миколаїв – 2024



<i>Ухань Є. О.</i> Використання пересувних джаммерів для побудови контрольованої зони.....	102
<i>Щекотов Р. В., Крайник Я. М.</i> Налаштування CI/CD-процесу розробки програмного забезпечення .....	104

### **Методи і засоби програмної інженерії**

<i>Андреев А. А., Кірей К. О.</i> Аналіз підходів до діагностування причин збоїв у мережі.....	105
<i>Бечка Д. Р., Давиденко Є. О.</i> Прогнозування ризиків в управлінні проєктами за допомогою логістичної регресії .....	108
<i>Глушко С. О.</i> Підвищення продуктивності розробки програмного забезпечення засобами штучного інтелекту.....	111
<i>Давиденко Є. О., Бондаренко С. В.</i> Інтелектуальне ресурсне планування в проєктному управлінні .....	113
<i>Жлуктарьов А. А., Давиденко Є. О.</i> Архітектурний стиль gRPC для високонавантажених систем.....	114
<i>Забєленков М. Д., Кандиба І. О.</i> <b>Методи та алгоритми розпізнавання та сегментації об'єктів на зображеннях за допомогою комп'ютерного зору.</b> .....	<b>116</b>
<i>Фінік В. Ю., Кандиба І. О.</i> Пошук дефектів на зображеннях, створених засобами штучного інтелекту.....	118

### **Вебтехнології та вебдизайн**

<i>Льчишина Ю. В., Швайко В. К.</i> Розробка дизайну інтерфейсу користувача мобільного застосунку спорт конект .....	120
<i>Кузьмін А. А.</i> Розробка дизайну інтерфейсу користувача мобільного застосунку еко-смартсїті .....	122
<i>Лопушанський К. А., Кандиба І. О.</i> Програмне забезпечення аналізу даних використання сайту з використанням методів штучного інтелекту .....	124

### **Інформаційні технології у навчальному процесі**

<i>Белоусова Я. Ю., Сіденко Є. В.</i> Інтелектуальна система обробки природної мови з використанням алгоритмів вебскрапінгу .....	127
---	-----

системами. Існує декілька стилів побудови API, але найбільш підходящий під контекст високонавантажених систем є gRPC.

Повідомлення в gRPC серіалізуються у вигляді високоєфективного бінарного формату даних Protobuf [2]. Порівняно з JSON він набагато швидше серіалізується та десеріалізується через невеликі та сильно стиснуті повідомлення. Особливий приріст в швидкодії та зменшенню витрат пам'яті видно при порівнянні швидкодії операцій десеріалізації.

Також, gRPC відрізняється більшою свободою в виборі методу обслуговування клієнтських застосунків: унарний, серверний потоковий, клієнтський потоковий та двухнаправлених потоковий.. Їх вдаль використання призводить до зростання швидкодії.

Отже, для критичних частин високонавантажених систем вкрай важливо знаходити шляхи отримання швидкодії навіть в таких фундаментальних рішеннях як вибір підходящого архітектурного стилю API.

#### **ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Somashekar G., Gandhi A. Towards Optimal Configuration of Microservice. EuroMLSys '21: Proceedings of the 1st Workshop on Machine Learning and Systems. 2021. pp 7-14. DOI: <https://doi.org/10.1145/3437984.3458828>.
2. Indrasiri K., Kuruppu K. gRPC: Up and Running. O'Reilly Media. 2020. 190 p.
3. Порівнюємо два формати серіалізації даних: Protobuf vs JSON. URL: <https://dou.ua/lenta/articles/protobuf-vs-json-go-benchmark/> (дата звернення: 08.01.24).

**УДК 004.4**

*Забеленков М. Д., Кандиба І. О.*

*Чорноморський національний університет ім. Петра Могили  
м. Миколаїв, Україна*

#### **МЕТОДИ ТА АЛГОРИТМИ РОЗПІЗНАВАННЯ ТА СЕГМЕНТАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНОГО ЗОРУ**

З розвитком штучного інтелекту та методів машинного навчання розпізнавання та сегментація об'єктів на зображеннях став актуальною галуззю у сучасному світі. Застосування методів та алгоритмів розповсюджені в різних галузях, а саме: медицина,

автономні транспортні засоби, відеоспостереження, робототехніка та тощо [1].

Розпізнавання об'єктів на зображеннях – це завдання в області комп'ютерного зору, яке полягає у визначенні та класифікації об'єктів або сутностей на зображенні [2]. До методів розпізнавання відносять глибокі нейронні мережі в основі яких є покладено згорткові та рекурентні. Згорткові використовуються для ефективного виявлення об'єктів та екстракції ознак на зображеннях. Рекурентні використовуються для врахування контексту та послідовності в завданнях, що потребують більш потужної обробки. R-CNN алгоритми – це один з найкращих методів розпізнавання не тільки об'єктів на статичних зображеннях, але й на відеоматеріалах у режимі реального часу.

Сегментація об'єктів на зображеннях – це завдання в області комп'ютерного зору, яке полягає у розділенні зображення на окремі регіони або сегменти. До методів сегментації відносяться наступні елементи. FCN та U-NET методи [3], що ефективно використовують контекст зображення для чіткої сегментації об'єктів на зображенні. GrabCut використовують чітко визначені користувачем області у яких сегментують зображення. Mask R-CNN є доповненням до одноіменного методу розпізнавання, яке дозволяє визначати та виділяти окремі об'єкти.

Різноманітні методи комп'ютерного зору дозволяють виконувати завдання різного типу та для різних цілей. Але насамперед кожен з цих методів обробки зображення можна групувати та отримувати більш розгорнутий аналіз наданої інформації.

Приклад використання алгоритмів та методів для сегментації та розпізнавання об'єктів:

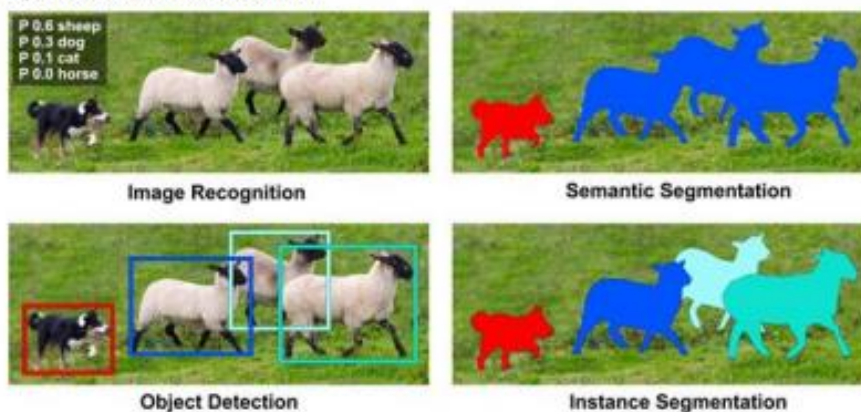


Рисунок 1 – Результат застосування методів та алгоритмів



Отже розпізнавання та сегментація об'єктів на зображеннях за допомогою комп'ютерного зору є важливими завданнями в області комп'ютерної обробки зображень та штучного інтелекту. Розвиток технології включає в себе подальше вдосконалення алгоритмів для роботи з різноманітними умовами та розширення його в нових галузях.

#### **ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Segmentation and object recognition using edge detection techniques / Y. Ramadevi, T. Sridevi, B. Poornima, B. Kalyani, international Journal of Computer Science & Information Technology (IJCSIT). Vol 2, No 6. 2010. P. 153-161.
2. Tetiana M., Kondratenko Y., Sidenko I. Computer Vision Mobile System for Education Using Augmented Reality Technology. Journal of Mobile Multimedia. 2021. P. 555–576.
3. Ozturk O., Sariturk B., Seker D. Z. Comparison of fully convolutional networks (FCN) and U-Net for road segmentation from high resolution imageries. International journal of environment and geoinformatics. Vol. 7, Issue 3. P. 272–279.

**УДК 004.8**

**Фіник В. Ю., Кандиба І. О.**

*Чорноморський національний університет імені Петра Могили  
м. Миколаїв, Україна*

#### **ПОШУК ДЕФЕКТІВ НА ЗОБРАЖЕННЯХ, СТВОРЕНИХ ЗАСОБАМИ ШТУЧНОГО ІНТЕЛЕКТУ**

Нейронні мережі стають доволі популярнішими, оскільки вони значно пришвидшують людську взаємодію з оточенням та інформацією. Їх можна класифікувати за різними критеріями та функціоналом і в цій роботі будуть розглянуті генеративні штучним інтелектом, а саме для створення різного роду медіа контенту. Їх можна використовувати для ілюстративних зображень, наприклад, коли використання справжніх людей може поставити їх під загрозу, для реклами та просування контенту, для відображення свого аватару в мережі та різного роду комерції. Це доволі перспективна галузь, оскільки хороших Prompt Engineer зараз не так багато.

Якщо ж говорити власне за нейромережі, то є декілька якими можна користуватись без складних маніпуляцій, а саме Midjourney (платна, підписка), DALL-E (інтегрована в ChatGPT-4, платна, система купівлі кредитів).

## ДОДАТОК Б

### Лістинг програми

```
import cv2
import numpy as np
import time
import torch
from sklearn.metrics import accuracy_score, confusion_matrix, precision_recall_fscore_support
# Зчитування зображення
image = cv2.imread('images/small.jpg', cv2.IMREAD_GRAYSCALE)

# Порогова бінаризація за допомогою методу Ніблека
def niblack_threshold(image):
    thresh_niblack = cv2.adaptiveThreshold(image, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
cv2.THRESH_BINARY, 11, -10)
    return thresh_niblack

# Порогова бінаризація за допомогою методу Оцу
def otsu_threshold(image):
    _, thresh_otsu = cv2.threshold(image, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    return thresh_otsu

# Балансування гистограми
def histogram_equalization(image):
    equalized_image = cv2.equalizeHist(image)
    return equalized_image

# Оцінка метрик
def evaluate_segmentation_metrics(ground_truth, segmented_image):
    pixel_accuracy = accuracy_score(ground_truth.flatten(), segmented_image.flatten())
    confusion_mat = confusion_matrix(ground_truth.flatten(), segmented_image.flatten())
    true_positive = np.diag(confusion_mat)
    false_positive = np.sum(confusion_mat, axis=0) - true_positive
    false_negative = np.sum(confusion_mat, axis=1) - true_positive
    true_negative = np.sum(confusion_mat) - (true_positive + false_positive + false_negative)
    mean_accuracy = np.mean(true_positive / (true_positive + false_positive + false_negative))
    mean_IoU = np.mean(true_positive / (true_positive + false_positive + false_negative))
    frequency_weighted_IU = np.sum(true_positive) / (
        np.sum(true_positive + false_positive) + np.sum(true_positive + false_negative))
    precision, recall, fscore, _ = precision_recall_fscore_support(ground_truth.flatten(), segmented_image.flatten(),
        average='weighted')
    mean_F1_score = np.mean(fscore)

    return pixel_accuracy, mean_accuracy, mean_IoU, frequency_weighted_IU, mean_F1_score

# Застосування методів сегментації та вимірювання часу
start_time = time.time()
thresh_niblack = niblack_threshold(image)
niblack_time = time.time() - start_time

start_time = time.time()
thresh_otsu = otsu_threshold(image)
otsu_time = time.time() - start_time
```

```
start_time = time.time()

UNET_time = time.time() - start_time

start_time = time.time()
equalized_image = histogram_equalization(image)
equalization_time = time.time() - start_time

# Оцінка метрик для кожного методуS
ground_truth = cv2.imread('images/small.jpg', cv2.IMREAD_GRAYSCALE)
metrics_niblack = evaluate_segmentation_metrics(ground_truth, thresh_niblack)
metrics_otsu = evaluate_segmentation_metrics(ground_truth, thresh_otsu)

metrics_equalization = evaluate_segmentation_metrics(ground_truth, equalized_image)

# Візуалізація результатів
Hori = np.concatenate((image, thresh_niblack, thresh_otsu, equalized_image), axis=1)
imS = cv2.resize(Hori, (1768,640))
cv2.imshow('Original, Niblack, Otsu, Histogram Equalization segmentation methods', imS)
cv2.waitKey(0)
cv2.destroyAllWindows()

model = torch.hub.load('yolov5', 'yolov5n', source='local')

img = cv2.imread('images/many.jpg')
img = cv2.resize(img, (1000,650))
result = model(img)
print(result)
df = result.pandas().xyxy[0]
print(df)

for ind in df.index:
    x1, y1 = int(df['xmin'][ind]), int(df['ymin'][ind])
    x2, y2 = int(df['xmax'][ind]), int(df['ymax'][ind])
    label = df['name'][ind]
    cv2.rectangle(img, (x1,y1), (x2,y2), (255,255,0), 2)
    cv2.putText(img, label, (x1,y1-5), cv2.FONT_HERSHEY_PLAIN, 2, (255,255,0), 2)

cv2.imshow('IMAGE',img)
cv2.waitKey(0)
# Виведення метрик для кожного методу
print('Metrics for Niblack Thresholding:')
print('Pixel accuracy:', metrics_niblack[0])
print('Mean accuracy:', metrics_niblack[1])
print('Mean IoU:', metrics_niblack[2])
print('Frequency weighted IU:', metrics_niblack[3])
print('Mean F1-score:', metrics_niblack[4])
print("")
print('Metrics for Otsu Thresholding:')
print('Pixel accuracy:', metrics_otsu[0])
print('Mean accuracy:', metrics_otsu[1])
print('Mean IoU:', metrics_otsu[2])
print('Frequency weighted IU:', metrics_otsu[3])
print('Mean F1-score:', metrics_otsu[4])
print("")
print('Metrics for Histogram Equalization:')
```

```
print('Pixel accuracy:', metrics_equalization[0])  
print('Mean accuracy:', metrics_equalization[1])  
print('Mean IoU:', metrics_equalization[2])  
print('Frequency weighted IU:', metrics_equalization[3])  
print('Mean F1-score:', metrics_equalization[4])
```