

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

(підпис) **В.О. РОМАНКЕВИЧ**
(ініціали, прізвище)

“ ____ ” червня 2021 р.

Дипломний робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою “Системне програмування та спеціалізовані
комп'ютерні системи”**

зі спеціальності **123 «Комп'ютерна інженерія»**

на тему: Комп'ютерні засоби розпізнавання об'єктів на зображенні

Виконав : студент IV курсу, групи КВ-73
Жовнірський Дмитро Миколайович

(підпис)

Керівник доц. каф. СПКС, к.т.н. Петрашенко А. В.

(підпис)

Консультант з нормоконтролю, доц.каф. СП і СКС, к.т.н. Клятченко Я.М.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент доц. каф. ПЗКС, к.т.н. Заболотня Т. М.

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2021 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Романкевич В.О.
(підпис) (ініціали, прізвище)
«__» червня 2021 р.

**ЗАВДАННЯ
на дипломний роботу студента**

Жовнірському Дмитру Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Комп'ютерні засоби розпізнавання об'єктів на зображенні»,
керівник роботи доц. каф. СПКС, к.т.н. Петрашенко А.В.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи: див. Технічне завдання

4. Зміст пояснювальної записки

- Аналіз існуючих рішень;
- Проблематика розпізнавання та методи обробки зображень;
- Порівняння інструментів реалізації;
- Архітектура розробленої програми.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

- Архітектура розробленої згорткової нейронної мережі. Схема алгоритму;

- Загальна схема проекту. Схема структурна;
- Блок-схема розробленої програми. Схема алгоритму;
- Схема виявлення об'єктів. Схема структурна.

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
нормоконтроль	к. т. н., доцент Клятченко Я.М.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Вивчення літератури за тематикою роботи	15.11.2020	
2	Розроблення та узгодження технічного завдання	30.11.2020	
3	Аналіз існуючих рішень	05.02.2021	
4	Підготовка матеріалів першого розділу дипломної роботи	07.05.2021	
5	Підготовка матеріалів другого розділу дипломної роботи	12.05.2021	
6	Підготовка матеріалів третього розділу дипломної роботи	14.05.2021	
7	Підготовка матеріалів четвертого розділу дипломної роботи	18.05.2021	
8	Підготовка графічної частини дипломної роботи	20.05.2021	
9	Оформлення документації дипломної роботи	23.05.2021	
10	Попередній захист дипломної роботи на кафедрі	26.05.2021	

Студент

(підпис)

Жовнірський Д.М.
(ініціали, прізвище)

Керівник роботи

(підпис)

Петрашенко А.В.
(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проекту.

АНОТАЦІЯ

Необхідність удосконалення розпізнавання об'єктів на зображенні комп'ютерними системами та широка різноманітність методів розпізнавання зображень зумовила напрям цієї роботи.

Мета дипломної роботи дослідити теоретичні відомості про принципи розпізнавання об'єктів на зображенні. На основі досліджень створити власну програму, яка може знаходити і розпізнавати об'єкти на зображенні.

В результаті досліджень в роботі було розглянуто основні принципи розпізнавання об'єктів на зображення, зроблено короткий огляд існуючих методів вирішення задач розпізнавання об'єктів на зображенні.

Було описано їх основні переваги та недоліки. В роботі було розглянуто обробки зображень. Розроблено алгоритм та архітектуру програми розпізнавання об'єктів на зображенні.

Проведено тестування для демонстрації роботи програми. Програма здатна з точністю обробляти об'єкти на зображенні різної якості, виділяти їх розташування, класифікувати та розпізнавати їх.

Дипломна робота включає у себе пояснювальну записку (59 ст., 27 рис., 1 табл., 2 додатки).

Ключові слова: розпізнавання об'єктів, обробка зображень, YOLOv3, нейронна мережа, C#, CNN.

SUMMARY

The need to improve the recognition of objects in the image by computer systems and the wide variety of methods of image recognition has determined the direction of this work.

The purpose of the thesis is to explore theoretical information about the principles of object recognition in the image. Based on research, create your own program that can find and recognize objects in the image.

As a result of research, the basic principles of image recognition on the image were considered, a brief overview of the existing methods of solving the problems of object recognition on the image was made.

Their main advantages and disadvantages were described. Image processing was spilled in the work. The algorithm and architecture of the program for recognizing objects in the image have been developed.

Testing was conducted to demonstrate the operation of the program. The program is able to accurately process objects in the image of different quality, highlight their location, classify and recognize them.

Thesis includes an explanatory note (59 pages, 27 pictures, 1 table, 2 appendices).

Keywords: object recognition, image processing, Yolov3, neural network, C#, CNN.

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
1.1. Вимоги до програмного продукту, що розробляється	2
1.2. Вимоги до апаратного забезпечення.....	2
1.3. Вимоги до програмного та апаратного забезпечення користувача	2
6. ЕТАПИ РОЗРОБКИ	3

					ІАЛЦ.045470.002 ТЗ			
Зм	Лист	№ докум.	Підп.	Дата	Комп'ютерні засоби розпізнавання об'єктів на зображенні. Технічне завдання	Літ.	Лист	Листів
Розроб.		Жовнірський Д.М.						
Перев.		Петрашенко А.В.					1	3
Н. контр.		Клятченко Я.М.				НТУУ «КПІ ім. Ігоря Сікорського», ФПМ, КВ-73		
Затв.		Романкевич В.О.						

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Комп'ютерні засоби розпізнавання об'єктів на зображенні».

Галузь застосування: системи досліджень, системи розвідки.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломну роботу на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даної роботи є дослідження програмних засобів для розпізнавання об'єктів на зображенні та створення власної програми для розпізнавання.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

1.1. Вимоги до програмного продукту, що розробляється

- кросплатформеність;
- гнучкість реалізації;
- наявність виводу оброблених зображень;
- наявність вибору файлів.

1.2. Вимоги до апаратного забезпечення

- Оперативна пам'ять: 4 Гб;
- Наявність доступу до відео камери.

1.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows, Linux;

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	25.02.2021
2.	Розроблення та узгодження технічного завдання	12.03.2021
3.	Аналіз існуючих рішень	15.03.2021
4.	Розробка програми	20.04.2021
5.	Тестування	03.05.2021
6.	Розробка графічної частини дипломної роботи	04.05.2021
7.	Підготовка матеріалів для дипломної роботи	06.05.2021
8.	Оформлення документації дипломного проєкту	07.05.2021
9.	Підготовка матеріалів графічної частини проєкту	15.05.2021

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4	ІАЛЦ.045490.000	Завдання на дипломну роботу	2	
2	A4	ІАЛЦ.045490.001 ОА	Опис альбому	2	
3	A4	ІАЛЦ.045490.002 ТЗ	Технічне завдання	3	
4	A4	ІАЛЦ.045490.003 ТП	Відомість технічного проєкту	1	
5	A4	ІАЛЦ.045490.004 ПЗ	Пояснювальна записка	59	
6	A4	ІАЛЦ.045490.005 Д1	Структурна схема засобу. Схема структурна	1	
7	A4	ІАЛЦ.045490.006 Д2	Архітектура розробленої згорткової нейронної мережі. Схема структурна	1	
8	A4	ІАЛЦ.045490.007 Д3	Загальна схема роботи. Схема алгоритму	1	
9	A4	ІАЛЦ.045490.008 Д4	Блок-схема розробленої програми розпізнавання об'єктів. Схема алгоритму	1	
				ІАЛЦ.045490.003 ТП	
		ПІБ	Підп.	Дата	
Розробн.	Жовнірський Д.М.				Лист
Керівн.	Перащенко А.В.				Листів
Консульт.					1
Н/контр.	Клятченко Я.М.				1
Зав.каф.	Романкевич В.О.				КПІ ім. Ігоря Сікорського Каф. СПіСКС Гр. КВ-73

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	4
ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	7
1.1. Нейронні мережі.....	7
1.1.1. Проста нейронна мережа	7
1.1.2. Згортоква нейронна мережа	9
1.2. Віюли Джонса.....	13
1.3. Шаблонний метод	15
1.3.1. Проста відповідність.....	15
1.3.2. Відповідність на базі характерних рис	15
1.3.3. Відповідність на базі областей.....	16
1.3.4. Кореляція зображень.....	16
1.4. Детектор точок	17
1.5. Контурний аналіз.....	19
1.6. Висновки	22
РОЗДІЛ 2. ПРОБЛЕМАТИКА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ТА МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ	24
2.1. Проблематика розпізнавання об'єктів	24
2.2. Елементарні методи обробки зображень	25
2.2.1. Попередня обробка	26
2.2.2. Сегментація	26
2.2.2.1. Морфологічні методи	27
2.2.2.2. Пороговий метод	27
2.2.2.3. Метод нарощування областей	28

					ІАЛЦ.045490.004 ПЗ			
Зм	Лист	№ докум.	Підп.	Дата	Комп'ютерні засоби розпізнавання об'єктів на зображенні Пояснювальна записка	Лім.	Лист	Листів
Розроб.		Жовнірський Д.М.						
Перев.		Петрашенко А.В.					1	61
Н. контр.		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ, КВ-73		
Затв.		Романкевич В.О.						

2.2.2.4. Текстурний метод	29
2.2.2.5. Контурна сегментація	30
2.2.3. Фільтрація	31
2.2.4. Розпізнавання об'єктів	32
2.3. Висновки	33
РОЗДІЛ 3. ПОРІВНЯННЯ ІНСТРУМЕНТІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМИ....	34
3.1. Мови програмування	34
3.1.1. C++	34
3.1.2. Python	35
3.1.3. C#	36
3.2. Бібліотеки для реалізації	37
3.2.1. OpenCV	37
3.2.2. TensorFlow	39
3.2.3. YoLov3	41
3.4. Порівняння бібліотек	43
3.5. Висновки	45
РОЗДІЛ 4. АРХІТЕКТУРА РОЗРОБЛЕНОЇ ПРОГРАМИ.....	46
4.1. Навчання нейронної мережі	46
4.2. Архітектура нейронної мережі	47
4.3. Розроблення програми	49
4.4. Тестування	50
4.4. Висновок	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	58

ДОДАТКИ

Додаток 1. Копії графічних матеріалів

- ІАЛЦ.045490.005 Д1. Алгоритм тренування нейронної мережі . Схема алгоритму;
- ІАЛЦ.045490.007 Д2. Архітектура розробленої згорткової нейронної мережі. Схема алгоритму;
- ІАЛЦ.045490.006 Д3. Загальна схема роботи. Схема алгоритму;
- ІАЛЦ.045490.008 Д4. Блок-схема розробленої програми розпізнавання об'єктів. Схема алгоритму.

Додаток 2. Фрагменти програмного коду

					ІАЛЦ.045490.004 ПЗ	Лист
						3
Зм	Лист	№ докум.	Підп.	Дата		

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

API – набір інструментів підпрограм, які складаються з протоколів взаємодії та засобів для створення програмного забезпечення.

CLR - це набір служб, необхідних для виконання керування коду.

CNN – згорткова нейронна мережа. Використовується для аналізу зображення, як одна із найкращих глибинних нейронних мереж.

FCL - це об'єктно-орієнтована бібліотека класів, інтерфейсів і системи типів, які входять до складу Microsoft .NET.

FrameworkNet - технологія, яка підтримує створені та виконані програми Windows.

Open Source – відкрите програмне забезпечення. Вільний доступ.

SQL – деклативна мова програмування, яка використовується в роботі з таблицями даних.

					ІАЛЦ.045490.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		4

ВСТУП

В час автоматизації і роботизації процесів у промисловості, науці, а також побуті приділяється велика увага у розробленні інтелектуальних інформаційних систем.

Інформаційні технології використовують в різних галузях:

- медична діагностика;
- сільське господарство;
- ядерна та космічна наука;
- лінгвістика;
- автоматизовані системи управління;
- криміналістика.

Сфера застосування даних систем в край велика і продовжує збільшуватися з кожним днем на хвилях технічного прогресу.

Тому теорія і технологія створення програм та машин, для розпізнавання і класифікації об'єктів на зображенні дуже важлива.

Основним змістом розпізнавання об'єктів є здобуття інформації із зображень. Тому і тема, роботи виявлення і розпізнавання об'єктів є актуальною.

Протягом досить тривалого часу проблема розпізнавання образів привертає увагу науковців. Так у 20-х роках роботи І. Фішера призвела до формування дискримінантного аналізу, як одного із розділів теорії і практики розпізнавання.

У 40-х роках науковці А. М. Калмогоров і О. Я Хінчин поставили задачу про розділення суміші двох розподілів.

У 50-60 роках з'являється теорія статистичних рішень.

В результаті великої кількості робіт було відкрито алгоритми, які порівнюють новий об'єкт з іншим у якого вже заданий клас, що стало початком планомірного наукового пошуку і практичних розробок.

					ІАЛЦ.045490.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		5

В рамках кібернетики сформувався новий науковий напрям, який розробляв теорію і практичну реалізацію пристроїв, а потім і системи, які були призначені для розпізнавання об'єктів, явищ, процесів. Нова наукова дисципліна отримала назву «Розпізнавання образів».

В 1952 р. – був створений перший пристрій розпізнавання мовлення. В 1957 р. – перцептрон Розенблатта. В 1969 р. – в електротехнічній лабораторії Японії почалась розробка проекту «промисловий інтелектуальний робот»

1970-х роках – розпізнавання формується, як самостійний науковий напрям. Значний внесок у розвиток системи та засобів інтелекту зробили вітчизняні та зарубіжні вчені: М. Ф. Бондаренко, В. І. Васильєв, Т. К. Вінцюк, В. В. Грицик, М. В. Завалишин, І. Б. Мучник, Є. П. Путятін, Б. П. Русін, М. І Шлезінгер.

Розпізнавання образів має широке застосування і використовується при створенні усіх систем, на які покладаються інтелектуальні функції.

В сучасному світі все частіше і актуальніше стає питання розробки і застосування технологій машинного інтелекту для вирішення численних задач.

Тому, актуальність вивчення теми розпізнавання образів зумовлена потребою у розробленні інтелектуальних інформаційних систем.

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Дослідження, розробка, реалізація методів розпізнавання та оцінювання об'єктів на зображенні є провідним напрямом у розвитку сучасного програмного забезпечення. Розпізнавання образів в останні роки знаходить все більше застосування.

Здатність сприймати зовнішній світ у формі образів дозволяє дослідити властивості об'єктів та моделювати процес їх розпізнавання.

В даний час розроблено досить багато різноманітних методів розпізнавання об'єктів на зображенні кожен з яких був створений для роботи з конкретним типом зображень. З точки зору конкретного завдання потрібно обирати найбільш оптимальний і досконалий метод.

В цьому розділі розглянуті деякі з існуючих методів розпізнавання об'єктів на зображенні.

1.1. Нейронні мережі

1.1.1. Проста нейронна мережа

Нейронна мережа є високоякісною сучасною технологією, в основі якої лежить робота людського мозку.

Основою людського мозку є нейрони. Складовою штучної нейронної мережі є перцептрон. Основою нейронних мереж є штучний нейрон модель якого зображені на рис. 1.1.

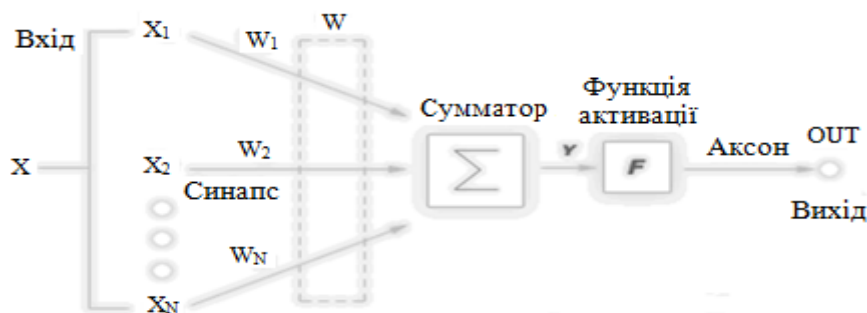


Рисунок 1.1 – Модель штучного нейрону

Існує багато видів штучних нейронних мереж і кожна унікальна і особлива.

Штучний нейрон – нейронна мережа, що рухається. У цій нейронній мережі дані рухаються в одному напрямку, вони не мають зворотнього розповсюдження. Нейронна мережа може мати 1 шар, або приховані шари [1]. На рис 1.2 приклад одношарової нейронної мережі, що рухається.

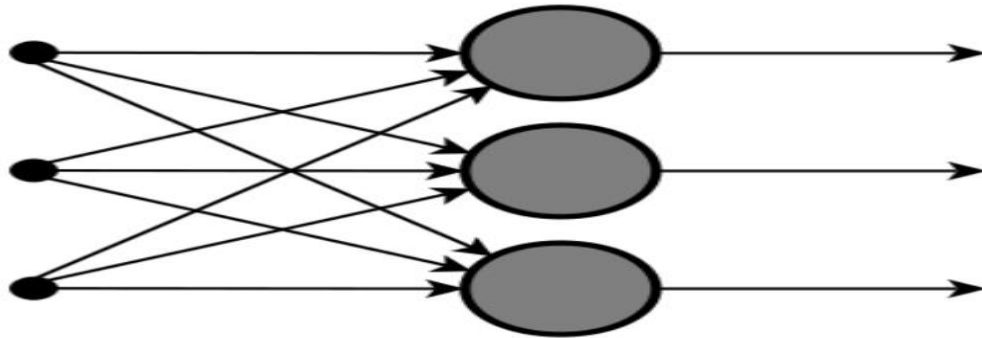


Рисунок 1.2 - Одношарова нейронна мережа, що рухається.

Ці нейронні мережі використовують для розпізнавання об'єктів, облич на зображенні.

Переваги простої нейронної мережі в тому, що вона досить проста в обслуговуванні та їх краще використовувати для обробки даних, які містять шум.

Багатошаровий перцептрон має 3 або більше шарів. Приклад багатошарового перцептрону зображено на рис. 1.3.

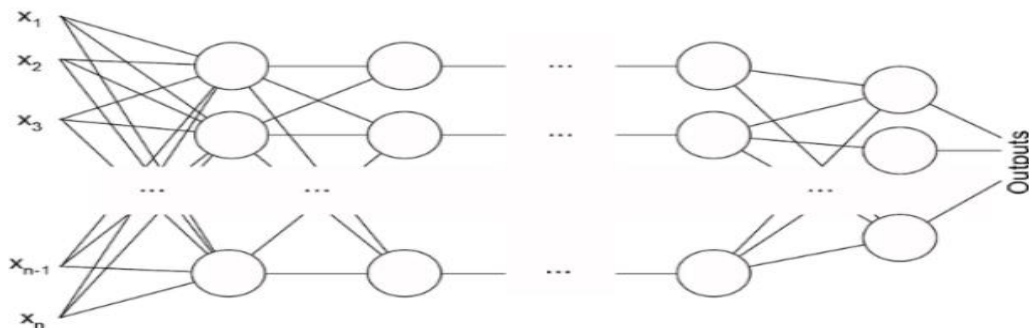


Рисунок 1.3 - Багатошаровий перцептрон

1.1.2. Згорткова нейронна мережа

Ефективні результати в розпізнаванні об'єктів на зображенні та відео дають згорткові нейронні мережі (CNN).

Для організації згорткової нейронної мережі застосовується 3 шари:

- згортки;
- підвибіки;
- повнозв'язний шар.

Першим кроком є вхідні данні "Input", які містять інформацію про зображення (ширина 128, висота 128, 3 канали кольору Red, Green, Blue) . Наступний крок є шар згортки "Convolution", де відбувається поелементне множення після чого всі ці множення підсумовуються і записуються у нову структуру даних [5].

Згорткова нейронна мережа має велику кількість фільтрів завдяки яким відбувається пошук характеристик. Кожна унікальна позиція введеного зображення виробляє число, яке записується у структуру даних. Якщо використовують 5 фільтрів данні зміняться на $128 \times 128 \times 5$ (рис.1.4).

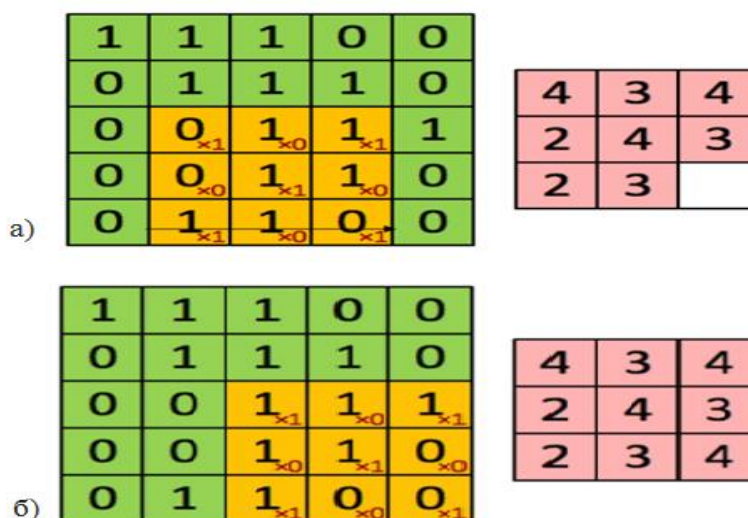


Рисунок 1.4 – а) фільтр на одному з каналів зображення;

б) результат накладання фільтру.

Наступний крок “Pool” - зменшення розміру зображень (широти та висоти) $64 \times 64 \times 5$. Виконується нелінійне ущільнення картки ознак. Із частини пікселів ми обираємо ті, що мають найбільше значення.

Наступний крок блок нормалізації «Rectified linear units», анулює ознаки які не грають важливої ролі у розпізнаванні образів.

Наступний крок “Fully connected layer” виводить N-вимірний вектор (N - число класів) [3]. В цьому кроці дані із попереднього кроку зрівнюються з тими типами об’єктів, які нейронна мережа навчилася розпізнавати. Така структура дає змогу нейронній мережі розпізнавати різні об’єкти та показувати високий результат.

По суті кожен шар нейронної мережі використовує власне перетворення. В результаті такого опрацювання правильно класифікують картинку або потрібний об’єкт на зображенні. Приклади розпізнавання об’єктів за допомогою нейронних мереж показані на рис. 1.5.



Рисунок 1.5 Розпізнавання об’єктів нейронних мереж

Нейронні мережі широко використовують у всіх галузях. Нейронні мережі знайшли широке використання в медицині завдяки здатності класифікації об’єктів, наприклад в МРТ- діагностиці (рис.1.6).

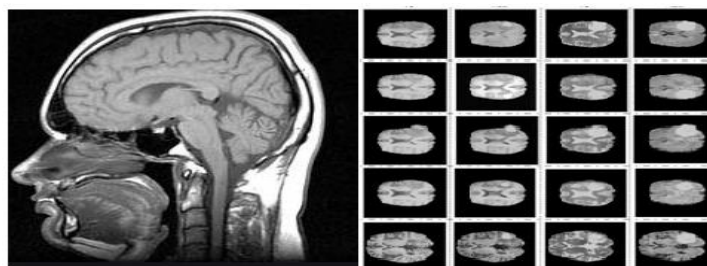


Рисунок.1.6 – Знімки МРТ

В агробізнесі даний метод використовують для прогнозування врожайності конкретних земель (рис.1.7).

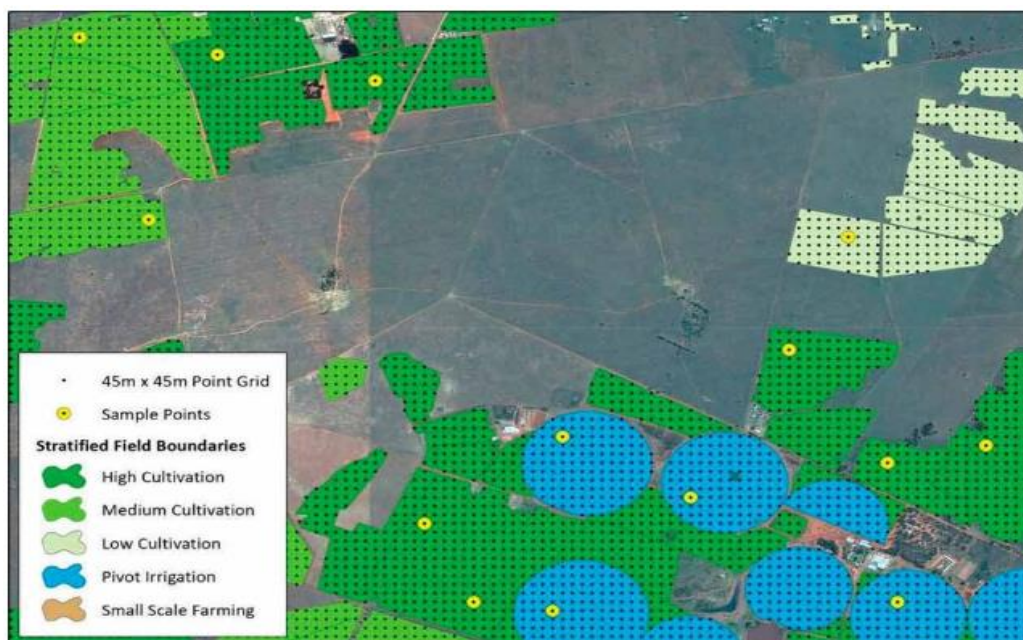


Рисунок 1.7 – Знімки спутнику з використанням розпізнавання нейронних мереж

За допомогою CNN можна розпізнавати обличчя конкретної людини (рис.1.8).



Рисунок 1.8 – Розпізнавання обличчя людей

Розпізнавання об'єктів на зображенні за допомогою нейронних мереж застосовують у відео спостереженні, безпілотному транспорті і так далі. Для відео аналітики також використовують нейронні мережі [6].

Крім реконструкції обличь (рис.1.9) згорткові мережі застосовують для реконструкції інших тривимірних об'єктів по зображенню.

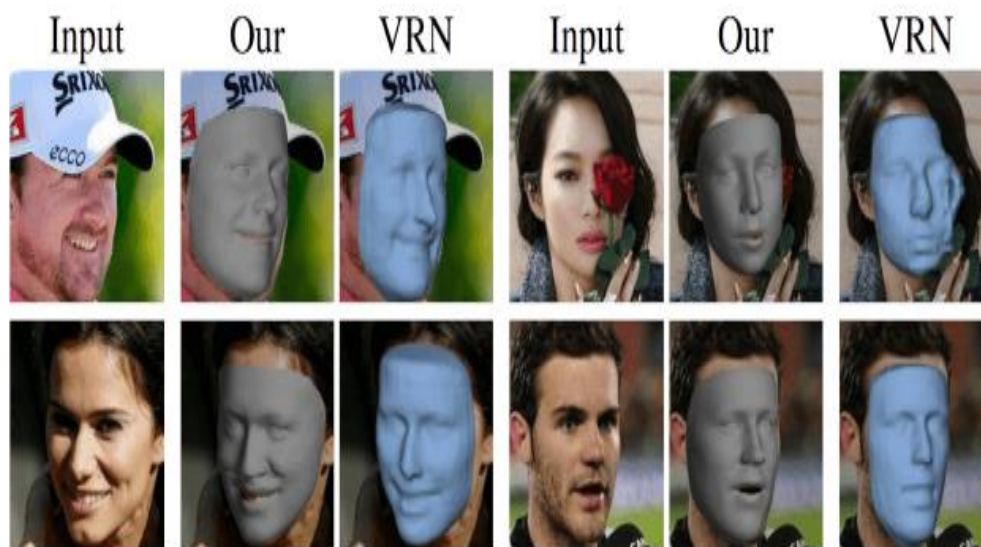


Рисунок 1.9 – Реконструкція тривимірних обличь.

Одна із сфер застосування розпізнавання об'єктів за допомогою нейронних мереж використовується для візуального пошуку. Суть цього методу полягає в тому, що людині достатньо сфотографувати предмет, або навести на нього камеру і програма за декілька секунд знайде цей товар і повідомить його характеристики та ціни.

Нейронні мережі є незамінними для вирішення деяких задач з якими не справляються або взагалі не можуть впоратися звичайні обчислювальні системи.

1.2. Віюли Джонса

На сьогоднішній день для розпізнавання об'єктів на зображенні найбільшого поширення отримав метод Віюли-Джонса в якому використовується принцип сканованого вікна. Цей метод запропонували Паул Віюла та Майкл Джонс у 2001 р. Пошук об'єктів здійснюється за допомогою ознак Хаара, основою для яких стали вейвлети Хаара [8].

Основні принципи, на яких базується метод, такі:

- інтегральне представлення зображень;
- використовуються ознаки Хаара;
- класифікатор оброблює всі вхідні дані та видає результат «вірно» або «брехня»;
- використовуються каскади ознак для швидкого відкидання вікон, в яких не було знайдено об'єкту.

Особливості, які використовували Віюла і Джонс є прямокутні хвилі однакової довжини, які представляють собою один високий інтервал і один низький інтервал. Прямокутна хвиля у двох вимірах являє собою пару сусідніх прямокутників - один світлий і один темний.

Прямокутні комбінації, краще підходять для візуального виявлення об'єкта (рис.1.10).

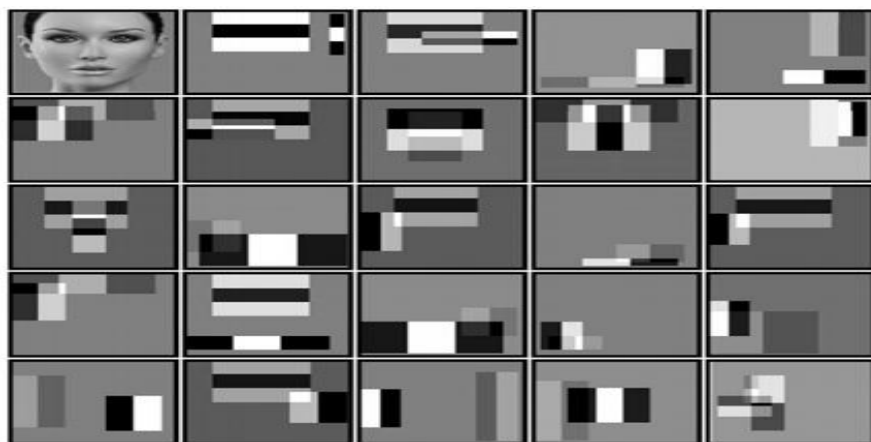


Рисунок 1.10 - Прямокутні комбінації візуального виявлення

Наявність функції Хаара визначається за допомогою вирахування середнього значення області світлих пікселів з середнього значення області темних пікселів (1):

$$F_{\text{haar}} = R_{\text{white}} - R_{\text{black}}. \quad (1)$$

Якщо різниця перевищує поріг, тоді говорять, що функція існує.

Віола та Джонс використали метод інтегрального представлення зображення.

Інтегральне представлення зображення - це матриця, що збігається за розмірами з вихідним зображенням. Значення кожного пікселя є сума значень всіх пікселів над ним та зліва від нього. Елементи матриці розраховуються за формулою (2):

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j), \quad (2)$$

де $I(i, j)$ - яскравість пікселя вихідного зображення.

Кожен елемент матриці $L(x, y)$ є сумою пікселів в прямокутнику від $(0,0)$ до (x, y) , тобто значення кожного пікселя (x, y) дорівнює сумі значень усіх пікселів лівіше і вище даного пікселя (x, y) . Інтегральне зображення прораховується за один прохід, тому що розрахунок матриці займає лінійний час, пропорційний числу пікселів в зображенні [5].

Розрахунок матриці можливий за формулою (3):

$$L(x, y) = I(x, y) - L(x-1, y-1) + L(x, y-1) + L(x-1, y). \quad (3)$$

За такою формулою можна швидко вирахувати суму пікселів довільного прямокутника, довільної площі (рис 1.11).

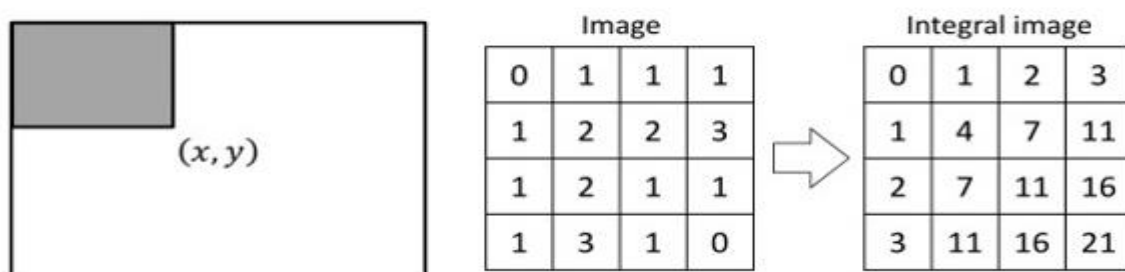


Рисунок 1.11 – Приклад швидкого вирахування пікселів площі

Сама найпоширеніша реалізація цього метода належить бібліотеці комп'ютерного зору OpenCV . На даний час існує кілька оптимізацій метода Віюлі-Джонса, що дає можливість підвищити достовірність розпізнавання об'єкту та прискорити швидкість знаходження.

1.3. Шаблонний метод

Шаблонний метод – це один із методів розпізнавання об'єктів на зображенні. Алгоритм роботи шаблонного методу спирається на зіставлення об'єкту зображення з ідеальним шаблоном. Цей метод використовується в медицині, виробництві в транспортній індустрії . Основні види розпізнавання об'єктів за шаблоном [2]:

1.3.1. Проста відповідність

Один із основних методів знаходження об'єкта на зображенні при пошуку за шаблоном – це проста відповідність. Суть методу полягає в покроковому розпізнаванні вхідного зображення з шаблоном. При кожному кроці розпізнавання вимірюється ступінь розпізнавання відповідності ділянки зображення шаблону. В кінці розпізнавання виділяються частина зображення, найбільш відповідна шаблону.

1.3.2. Відповідність на базі характерних рис

Цей метод використовується, коли зображення і шаблон мають більше співпадінь по характерними рисами та контрольними точками ніж цілим образом. Характерними рисами можуть бути точки, криві або моделі поверхні, які порівнюються на відповідність шаблону. Мета такого порівняння

знайти парні зв'язки між шаблоном та частиною зображення при цьому використовуються просторові співвідношення або характерні риси.

1.3.3. Відповідність на базі областей

Методи відповідності на базі областей або кореляційний метод, базується на комбінованому алгоритмі знаходжені характерних рис та відповідностей шаблонів.

В цьому методі порівняння відбувається на піксельному рівні, оскільки шаблони не мають помітних загальних характеристик зображення. За показниками інтенсивності шаблону та зображення встановлюється відповідність. Бувають випадки коли відповідність між шаблоном і зображенням встановити неможливо. В таких випадках при знаходженні відповідності використовують власне значення та власний простір. Вони містять інформацію, яка потрібна для порівняння образів при різних умовах освітлення і контрастності контуру.

1.3.4. Кореляція зображень

У цьому методі початкове зображення і шаблон можуть мати різні інтенсивності зображення. Порівняння проводиться по рівню подібності на основі кореляцій між шаблоном та оригіналом.

До переваг шаблонного методу відносяться: простота реалізації; надійна робота в умовах відсутності перешкод; висока точність розпізнавання; швидкість.

До недоліків можна віднести: сильна залежність від шаблонів і складність підбору оптимальних шаблонів; повільна робота при великій кількості перешкод; чутливість до обертання, шумів і спотворень.

1.4. Детектор точок

Для кожного об'єкта на зображенні можна знайти особливі точки інтересу і описувати ними сам об'єкт. Процес визначення особливих точок досягається шляхом використання детектора та дескриптора.

Детектор - це метод вилучення особливих точок з зображення.

Дескриптор – ідентифікатор особливої точки, що виділяє її з іншої безлічі особливих точок.

У 1992 Haralick і Shapir виділили такі вимоги до особливих точок [9]:

- відмінність - особлива точка повинна виділятися на тлі і бути унікальною;
- інваріантність - визначення особливої точки повинно бути незалежним до афінних перетворень;
- стабільність - визначення особливої точки повинно бути стійким до шумів і помилок;
- унікальність - особлива точка повинна бути унікальною для поліпшення розрізнення повторюваних патернів;
- інтерпретованість - особливі точки повинні визначатися так, щоб їх можна було використовувати для аналізу відповідностей з зображенням.

Tuytelaars і Mikolajczyk (2006) доповнили список властивостей, якими повинні володіти особливі точки:

- повторюваність;
- відмінність;
- локальність;
- кількість;
- точність;
- ефективність.

Одним з найперших детекторів особливих точок зображення був запропонований оператор Моравека. Він визначає зміни зображення блока 4 на 4 пікселі у горизонтальному, вертикальному, діагональному та протидіагональному напрямках (рис 1.12). Сила кута вираховується, як найменша різниця квадратів між полем і його сусідами [4].

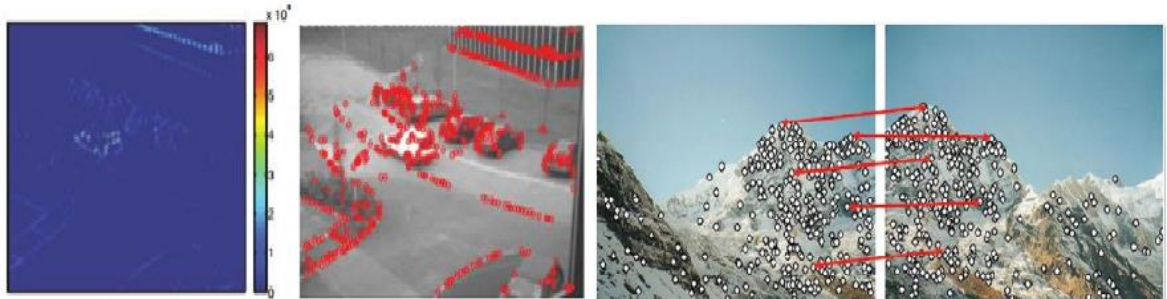


Рисунок 1.12 – Особливі точки за допомогою оператора Морвека

Поліпшення методу Моравець є детектор кутів Харріса. Метод Харріса інваріантний до обертання, а також до афінних змін яскравості, однак він чутливий до шумів та залежить від масштабів зображення. По результатам досліджень Він є одним із найоптимальніших детекторів кутів.

Детектор Харріса розглядає зображення, як функцію двох змінних I_{xy} . Для кожного пікселя зображення обчислюється значення особливої функції відгуку кута. Для цього розраховується матриця Харріса (4-6).

$$R = \det M - k(\text{tr} M)^2 > k, \quad (4)$$

$$M = \sum_{(u,v) \in w} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \quad (5)$$

$$R = \det M - k(\text{tr} M)^2 > k, \quad (6)$$

R – міра кута, де k – емпірична константа.

Shi-Tomasi – використовується та ж сама матриця M . Але визначаються мінімальні власні числа матриці напрямку. Робиться припущення, що при використанні цього алгоритму пошук кутів буде більш стабільним.

SIFT- будуються піраміди Гаусіан і визначаються екстремуми. Далі фільтруються особливі точки, визначається орієнтація ключових точок і будуються дескриптори.

1.5. Контурний аналіз

Контурний аналіз дає змогу розпізнавати об'єкти на зображенні. Передбачають, що контур містить всю необхідну інформацію про форму об'єкта. До уваги не беруться внутрішні точки. Контурний аналіз дає змогу зменшити обчислювальну і алгоритмічну складність [1].

Методи контурного аналізу актуальні у сфері розпізнавання об'єктів. Ці методи використовують у різних галузях. Усі методи контурного аналізу базуються на властивості яскравості – розривності. Для пошуку цієї властивості використовують ковзну маску, що є матрицею коефіцієнтів відповідно до пікселів вхідного зображення. Цю матрицю використовують для виконання просторової фільтрації. Градієнт рівня яскравості обраховується у кожній точці. Якщо зображення однотонного тростору градієнт рівня яскравості буде низький. У вихідному положені такі простори зображення темнішають. Якщо рівень градієнта високий на вихідному зображенні простори будуть яскраві.

Спочатку знаходять R (рівень фільтрації) (7):

$$R = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 0)f(x + 1, y) + w(1, 1)f(x + 1, y + 1), \quad (7)$$

тобто це сума фільтрації до бутку коефіцієнтів маски на значення пікселів, де була застосована маска.

Градієнт яскравості визначається за дискретними аналогами похідних першого і другого порядку.

Перша похідна $f(x)$ – це різниця значень сусідніх пікселів. Друга похідна – різниця значень сусідніх значень першої похідної (8):

$$\frac{\delta f}{\delta x} = f(x+1) - f(x), \frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x). \quad (8)$$

У випадку двох змінних (x, y) обчислюється часткові похідні за двома просторовими осями.

Перша похідна зображення обчислюється на дискретних наближеннях двовимірного градієнта.

Градієнт зображення це вектор (9):

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}. \quad (9)$$

Важливою частиною у методах контурного аналізу є модуль вектора градієнта функції (10):

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}. \quad (10)$$

Знаходять напрямок вектора градієнта (11):

$$a(x, y) = \arctg\left(\frac{G_y}{G_x}\right). \quad (11)$$

Напрямок контуру (x,y) буде перпендикулярним напрямку вектора градієнта.

Для знаходження контурів об'єктів на зображенні використовують метод лінійної фільтрації.

У цьому випадку використовують такі фільтри:

- різницевий амплітудний фільтр (12)

$$y_{ij} = \frac{1}{2} \sqrt{[(a_{i,j} - a_{i,j+1}) + (a_{i+1,j} - a_{i+1,j+1})]^2 + [(a_{i,j} - a_{j+1,i}) + (a_{i,j+1} - a_{i+1,j+1})]^2} \quad (12)$$

- максимальний різницевий амплітудний фільтр (13)

$$y_{ij} = |\max(a_{i,j}; a_{i,j+1}; a_{i+1,j}; a_{i+1,j+1}) - \min(a_{i,j}; a_{i,j+1}; a_{i+1,j}; a_{i+1,j+1})| \quad (13)$$

- фільтр Робертса (14)

$$y_{i,j} = \sqrt{(a_{i,j} - a_{i+1,j+1})^2 + (a_{j+1,j} - a_{i,j+1})^2} \quad (14)$$

- фільтр Прюїтта (15)

$$y_{ij} = \frac{1}{2} \sqrt{[(a_{i,j} - a_{i,j+2}) + (a_{i+1,j} - a_{i+1,j+2}) + (a_{i+2,j} - a_{i+2,j+2})]^2 + [(a_{i,j} - a_{i+2,j}) + (a_{i,j+1} - a_{i+2,j+1}) + (a_{i,j+2} - a_{i+2,j+2})]^2} \quad (15)$$

- фільтр Собеля (16)

$$y_{ij} = \sqrt{[(a_{i,j} - a_{i,j+2}) + 2 \cdot (a_{i+1,j} - a_{i+1,j+2}) + (a_{i+2,j} - a_{i+2,j+2})]^2 + [(a_{i,j} - a_{i+2,j}) + 2 \cdot (a_{i,j+1} - a_{i+2,j+1}) + (a_{i,j+2} - a_{i+2,j+2})]^2} \quad (16)$$

Маски таких фільтрів для вікна розміром 3×3 мають такий вигляд:

а) Робертса:

Для осі ОХ

0	0	0
0	-1	0
0	0	1

Для осі ОУ

0	0	0
0	0	-1
0	1	0

б) Собеля:

Для осі ОУ

-1	-2	-1
0	0	0
1	2	1

Для осі ОХ

-1	0	1
-2	0	2
-1	0	1

в) Прюїтта:

Для осі ОХ

-1	-1	-1
0	0	0
1	1	1

Для осі ОУ

-1	0	1
-1	0	1
-1	0	1

Оператор Прюїтта підходить для роботи з зображеннями з високою загальною яскравістю і він є найменш ктратний. За найнижчою загальною яскравістю зображень для роботи підходить оператор Робертса, але він найбільш затратний метод. Оператор Собеля виконується швидше і є оптимальним щодо витрат [2].

Для знаходження контуру використовую ланцюговий код Фрімана. При використанні кода Фрімана будується матриця напрямків, або матриця зв'язності центральної точки з усіма сусідніми точками (рис.1.13).

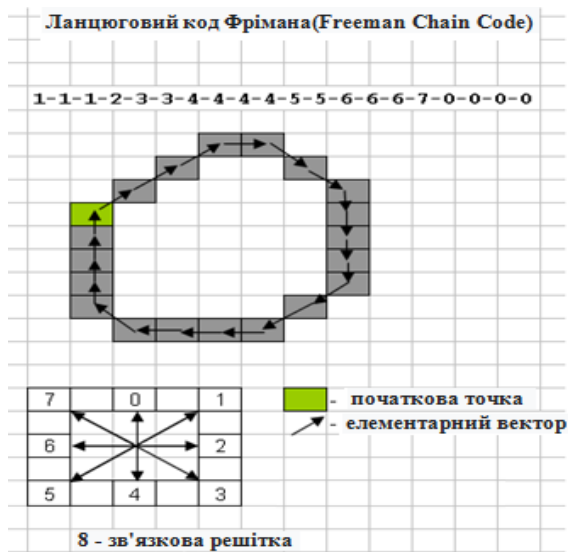


Рисунок 1.13 – Ланцюговий код Фрімана

Кожне з восьми напрямків переходу від точки до точки кодується трехбітним кодом. Задається початкова точка з неї починається виділення контуру.

Контурний аналіз дозволяє описувати, зберігати, порівнювати і розпізнавати об'єкти представлені у вигляді своїх зовнішніх обрисів – контурів. Контурний аналіз допомагає вирішити основні проблеми розпізнавання образів – перенесення, поворот, зміна масштабу зображення об'єкти.

1.6 Висновок

В цьому розділі були розглянуті методи, які використовують для розпізнавання об'єкта на зображенні.

Існує велика кількість методів розпізнавання об'єктів, але універсального методу для всіх умов розпізнавання не існує. Вибір конкретного методу залежить від багатьох умов.

Якщо об'єкт складний, але добре виділяється корисними будуть методи виділення країв та контурний аналіз.

Метод співставлення з шаблоном буде мало ефективним, якщо зображення об'єкта повернуте чи масштабне відносно шаблону. В цьому випадку краще застосовувати методи засновані на особливих точках. Вони дозволяють співставити об'єкт зі схожими класами об'єктів, або співставити об'єкт сам з собою.

Для розпізнавання об'єктів певного класу ми можемо використовувати, як метод позитивних і негативних наборів зображень, так і виділення кластерів дескрипторів особливих точок і створення словника дескрипторів.

Ефективними розпізнавання образів на зображення є нейромережеві методи. Але вони вимагають великої кількості обчислювальних ресурсів та часу, а це ускладнює їх застосування, коли потрібно швидко розпізнати об'єкт.

Отже, задача розпізнавання об'єктів може бути вирішена лише за конкретних умов.

І тому на сьогодні дуже важливою і актуальною є розробка таких методів розпізнавання об'єктів на зображенні, що можуть швидко і ефективно обробляти образи.

Сфера застосування даних технологій вкрай велика і продовжує збільшуватися з кожним днем на хвилях технологічного прогресу.

РОЗДІЛ 2. ПРОБЛЕМАТИКА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ТА МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ

2.1. Проблематика розпізнавання об'єктів

Основне завдання розпізнавання зрозуміти, чи стосуються данні на зображенні шуканому об'єкті. Необхідно не просто виділити об'єкт, але й точно вказати в системі зображення його положення та розмір.

При розпізнаванні виникає ряд труднощів, які обумовлені такими чинниками:

- перешкоди та шум;
- складний текстурований фон, на якому відбувається виявлення об'єкту;
- ефекти загороджування одних об'єктів іншими об'єктами з невизначною формою;
- спотворюючі оптичні ефекти у вигляді різних розфокусувань, дисторсій об'єктивів, ракурсних спотворень і інші;
- різноманітність і змінність самих об'єктів. що призводить до змінної структури зображень;
- ефекти різкої зміни освітлення, відблиски, тіні;
- наявність дефектів та тимчасових змін форми об'єктів;
- зміна середовища розповсюдження світла між сенсорами та об'єктами спостереження (пил, дим, опади);
- несинхронність реєстрації та обробки даних, яка пов'язана з обмеженням швидкодії комп'ютерних засобів зберігання та аналізу зображень. А також збої в комп'ютерних програмах обробки.

Для вирішення задачі розпізнавання об'єктів на зображенні на різних етапах застосовують різні методи і прийоми. На рис. 2.1. представлені основні методи, що існують і застосовуються на сьогоднішній день.



Рисунок 2.1 – Основні методи розпізнавання об'єктів

2.2. Елементарні методи обробки зображень

Етап підготовки зображення до розпізнавання це поліпшення його візуальної якості: усунення розмитості, підвищення контрасту, підкреслення границь, фільтрація і т. д.

Задача розпізнавання комплексна і ділиться на етапи [2]:

- попередня обробка;
- сегментація;
- фільтрація;

- розпізнавання об'єктів.

Всі ці етапи використовуються при обробці зображень від початкового етапу до розпізнавання зображення.

2.2.1 Попередня обробка

Для підвищення ефективності та якості розпізнавання об'єктів на зображенні передуює попередня обробка зображень. Методи попередньої обробки залежать від мети розпізнавання і їх досить багато.

Основна мета попередньої обробки це зниження перешкод на зображенні, а також придушення зовнішніх шумів.

2.2.2 Сегментація

Сегментація – це поділ зображення на сегменти. Основна мета сегментації це спрощення зображення, щоб його легше було аналізу. Сегментація ґрунтується на двох принципах: розривності та подібності [7].

При принципі розривності основний підхід базується на визначенні контурів.

При принципі подібності орієнтуються на визначення порогового рівня та нарощування областей.

Результат сегментації є множина, яка покриває все зображення або множина виділена із зображення.

До основних методів сегментації відносяться такі методи:

- морфологічні методи;
- пороговий методи;
- методи нарощування областей;
- текстурні методи;
- контурна сегментація.

2.2.2.1. Морфологічні методи

Морфологічні методи – використовуються для роботи з чорнобілими зображеннями . Ці методи частіше використовують для подання форми об’єкту.

2.2.2.2. Пороговий метод

Порогова сегментація – особливість цього методу полягає в тому що вибирається порогове значення. Розділяють метод з оптимальним порогом і метод з адаптивним порогом. Всі інші методи є похідними від цих.

Частіше при обробці зображення використовують бінарізація.

Бінарізація - це процес перетворення кольорового зображення в чорнобіле. Матриця розміром $M \times N$ є пороговою поверхнею, що відповідає зображенню, яке отримуємо. Кожна клітина матриці задає поріг яскравості пікселю на вихідному зображенні. Яскравість кожного пікселя порівнюється з пороговим значенням яскравості. Якщо значення буде вище значення яскравості порогу, то на бінарному зображенні піксель буде “чорним” або “білим” [2].

У методах глобальної бінарізації порогова поверхня є постійним значенням порогової яскравості.

Методи локальної бінарізації змінюють значення порогової яскравості від точки до точки зображення, розрахунок проводиться в зоні локальних ознак в оточенні пікселя.

Основна мета бінарізації це зменшення кількості інформації. Існує велика кількість методів бінарізації. Кожен метод має свої переваги та недоліки. Так метод Отцу приводить до втрати дрібних деталей та “злипання об’єктів”, але цей метод швидкий в обробці і якість помилок в ньому менша ніж в інших методах.

Використовуючи метод Ніблика у випадках неоднорідності зображення з низькою контрастністю. Можуть з'явитися зайві об'єкти, але цей метод використовують для швидкої фільтрації контрастних зображень і при його використанні можна досягти великої швидкості обробки.

Метод Бернсена – самий швидкий серед інших методів, але при використанні цього методу для обробки однорідних зон яскравості з'являються помилкові чорні плями, що приводить до додаткової обробки – постпроцесинг.

Метод Еквеля підходить для використання обробки чітких і контрастних зображень. При обробці зображень з низькою контрастністю можуть виникати розриви та помилкові чорні області.

При обробці зображень яким властива нерівномірна яскравість використовують метод Яновіц і Брукштейна, але серед методів локальної бінарізації він найменш продуктивний.

Отже, можна зробити висновок, що кожний метод слід використовувати в конкретних випадках.

2.2.2.3. Методи нарощування областей

Метод нарощування областей – це поділ зображення на області, які мають подібні властивості.

Виділяють [7]:

- злиття з аналогічними областями;
- розщеплення області;
- комбінування двох методів;
- кластерний аналіз.

Метод нарощування областей використовують при обробці зображень на яких присутні шуми. Недоліком цього методу є те, що він виділяє загальні фрагменти, не показуючи змін яскравості всередині області.

В основі методу злиття з аналогічними областями лежить вибір центрів областей, до яких по черзі приєднуються сусідні точки, що відповідають певним критеріям. Процес зупиняється тоді, коли не одна точка неприєднується до області. Критерії на підставі яких точка приєднується або неприєднуються до області дуже різні: близькість до центру області, близькість за деякою статистикою, близькість до сусідньої точки і т. д.

Одним із основних методів розщеплення є метод водорозділу. Суть методу водорозділу полягає в тому, що спочатку будується зона контрасту зображення, а потім водорозділи (області високої контрастності). Отримані перегородки є лінії водорозділу.

Комбінування двох методів складається з розщеплення і злиття. Спочатку зображення розбивається на області. Процес розщеплення областей відбувається до тих пір, доки не отримано однорідності сегментів в розщепленні. Після цього іде процес злиття схожих сусідніх сегментів, поки не буде отримано розщеплення зображення на однорідні області максимального розміру.

Метод кластерного аналізу використовують для розщеплення зображення к-кластерів.

2.2.1.4. Текстурний метод

Текстура є яскравою ознакою сегментації. Один із способів текстурної сегментації є підрахунок міри зернистості текстури у всіх точках зображення і подальшою обробкою.

Вхідне зображення обробляють з метою перетворення його в зображення по яскравості.

Параметри текстури вимірюють в вікні, значення отримані в межах кордону між текстурними областями будуть усередненими. Тому важко точно визначити кордон між текстурними областями [6].

Інший підхід – це виявлення перехідної зони між областями з різною текстурою. Основна мета виявлення зміни текстури по яскравості. Відмінність між текстурними областями в усіх точках зображення посилюється, а потім піддається пороговому обмеженню для визначення текстурних перепадів. Параметри скануються по зображенню суміжних блоків розміри $W \times W$ елементів, знаходиться сума модулів перехресних різниць, після чого виконується порогове обмеження.

Задача розпізнавання текстур є перспективною і потрібною. Існує велика кількість методів для її розв’язання. Вони відрізняються кількістю елементів текстур, представленням елементів розміщення відносно сусідів і т. д. Потрібно обирати такі методи та підходи, які стануть складовою у розпізнаванні на зображенні.

2.2.1.5. Контурна сегментація

Одним із методів контурної сегментації – є метод виділення границь. Цей метод набув поширення в силу простоти свого використання.

Основою для побудови контурних методів є граничні детектори, які контролюють роботу граничних пікселів зображення по контрасту використовуючи маски.

Частіше використовують – симетричні маски, інваріантні маски повороту. Недоліком цього методу є зниження віддільності пікселів об’єкта від фону. Це іноді призводить до нестійкої сегментації. Для подолання цього використовують спрямовані маски.

Для виявлення границь, використовують маски, що створюються на перетворенні Фур’є. Їх використання складне і більш трудойомке.

Наступним кроком після виділення пікселів границі є зв’язування контурів і побудова границь зображення. В цьому випадку можуть з’явитися дві проблеми: поява розривів і потовщення границі.

Для подолання проблеми розриву границі використовують маски для зв'язування фрагментів границі, що лежать на одній лінії, методи гістерезису. Щоб побудувати межі зображення використовується пошук на графі [6].

До переваг контурних методів можна віднести простоту вирішення, стійкість до незначних перепадів яскравості і контрастності зображень. До недоліків складно усунення помилкових границь при низькій контрастності, а також складність при усуненні розривів границі при підвищеному порозі.

2.2.3. Фільтрація

Більшість зображень піддається впливу різного роду шумів. Наявність шумів на зображення може причинити неточності та спотворення, що приведе до ускладнення процесу розпізнавання зображення.

Фільтрація зображення проводиться з метою усунення різних шумів на зображенні та отриманні бажаного результату.

Виділяють частотну і просторову фільтрацію.

Частотні методи базуються на методі Фур'є. У випадках, якщо функція є періодичною уявлення називають рядами Фур'є. Непереодичну функцію, виражену у вигляді інтеграла від тригонометричних функцій, помножених на вагову функцію називають перетворенням Фур'є.

Просторова фільтрація застосовується для покращення зображень і задач відновлення зображень.

Покращення зображення – це надання зображенню форми, яка є придатна для подальшої обробки. Це видалення із зображення перешкод та шуму, підкреслення границь, задачі зміни контрасту.

Відновлення зображення – це отримання зображення, яке є близьким до ідеалу.

При рішенні цього завдання усувають шум, геометричні спотворення, розмиття, неоднорідність фону і т. д.

Фільтри, які застосовують це:

- фільтр, заснований на обчисленні середнього арифметичного;
- фільтр, заснований на обчисленні середнього геометричного;
- фільтр, заснований на обчисленні середнього гармонійного;
- фільтр, заснований на порядкових статистиках;
- федійний фільтр;
- фільтр, заснований на обчисленні максимуму і мінімуму;
- фільтр, заснований на виборі середньої точки.

Покращення якості зображення є важливою складовою нашого життя, і потребує постійного покращення та удосконалення методів вирішення цих завдань.

2.2.4 Розпізнавання об'єктів

Розпізнавання об'єктів – це кінцевий етап обробки зображень. Для вирішення задач розпізнавання використовують такі методи [7]:

- кореляційні методи – засновані на пошуку максимальної кореляції точок шаблону і областей зображення. Ці методи використовують при спостереженні за роботою машин, виявлення і розпізнаванні зображень в системі навігації, охоронній системі та системі розпізнання і т. д.;
- ознакові методи – це методи, які використовують виділення і зіставлення характерних ознак шуканого об'єкта з областю зображення, яка має ці ознаки. Основні проблеми при використанні цього методу – це вибір ознак;
- синтаксичний метод – заснований на отриманні структурно-лінгвістичних ознак. Правило з'єднання цих ознак, однакові для еталону і вхідного зображення;

- методи нормалізації – мета цих методів полягає в автоматичному обчисленні невідомих параметрів перетворень, якими є вхідні зображення і приведе їх до вигляду еталона;
- перетворення здійснюється за допомогою операторів нормалізації, а обчислення параметрів – функціоналами.

2.3. Висновок

Існує велика кількість методів обробки зображень. Що дає змогу поліпшити якість аналізу і прискорити його. Але не існує універсального методу обробки .

Тому доцільно використовувати методи під конкретні види зображень, що підвищить ефективність при обробці зображень.

РОЗДІЛ 3. ПОРІВНЯННЯ ІНСТРУМЕНТІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМИ

3.1. Мови програмування

3.1.1. C++

C++ є найбільш популярна серед поширених мов програмування, володіє одночасно як низькорівневою, так і високорівневою мовою програмування. Забезпечує більш високий рівень ефективності і контролю в контексті machine learning, ніж інші мови програмування.

Оскільки код, написаний на C++, займає значно менше місця ніж низькорівневі мови такі, як Assembler, C++ поширена в сферах розробки апаратного забезпечення. C++ підходить для реалізації алгоритмів у сфері ІТ. Більше ніж інші мови підходить для ресурсно містких додатків.

C++ об'єктивно-орієнтована, узагальнена та процедурна мова. Була розроблена у 1979 році Б'ярном Страуструпом в Bell Laboratories і названа "C з класами".

У 1983р. Страуструп перейменував мову у C++. В 1998 році мова C++ була стандартизована Міжнародною орган зацією стандартизації ISO/IEC. Стандарт C++ складається з 2 частин: ядра мови і стандартної бібліотеки. Стандартна бібліотека включає стандартну бібліотеку C з змінами, які більше відповідають мові C++. Інша частина бібліотеки відповідає Стандартній Бібліотеці Шаблонів. Вона містить визначення шаблонів контейнерів, ітераторів, алгоритмів. Нововведення, які були введені в C++ стосуються перетворень типів у вигляді функцій, розширення функцій, що підставляються: перевизначення; просторі імена; віртуальні функції; абстрактні класи і конструктори; перевизначення операторів; перевизначення шаблонів; обробка винятків; аргументи за умовчанням; підтримка об'єктно-орієнтовного програмування через класи; доповнення до типа даних.

До особливостей мови можна віднести, що мову C++ використовують коли треба дуже швидко обробити багато зображень її можна використовувати в додатку, написаному на іншій мові, наприклад Python.

До переваг мови можна віднести:

- програми на C++ мають досить хорошу швидкість;
- на мові C++ можна розробляти програми для різних платформ ;
- можна створити узагальнені алгоритми для різних типів даних;
- можна розробляти на низьких рівнях;
- має можливість використовувати SQL.

Мова C++ має свої недоліки для створення нового додатку. Потрібно писати великий об'єм коду, що займає багато часу і може викликати труднощі в обслуговуванні.

3.1.2. Python

Мова Python Foundation була створена в 1990-х роках і є високорівневою мовою програмування. Вона є однією із найбільш використовуваних мов для штучного інтелекту, широко використовується в науці. Python – відкрита мова, яка побудована на базі технології open source. Розробники весь час можуть доповнювати її.

Код інтерпретатора Python є відкрити, що дозволяє всім, хто зацікавлений брати участь у його розробці і поліпшенні.

До переваг мови можна віднести:

- гнучкість та простота;
- безліч фреймворків;
- можливість розширення;
- інтерпретируемість та малий час роботи;
- єдиний стандарт до написання коду;
- кросплатформеність та підтримання різних стилів та типів.

Такі переваги мови зробили її популярною. Python здатна вирішувати широке коло завдань і стати затребуваною на сьогоднішній день.

Мова Python розвивається і удосконалюється. Існує вже 3 версії мови. В грудні 2019 року перестала підтримуватися 2 версія мови.

Незважаючи на свої переваги, мова Python має свої недоліки:

- продуктивність;
- синтаксис;
- динамічна типізація;

Інтерпретатор Python використовує GIL, а це ускладнює процес паралельних обчислень. Але мова Python весь час розвивається та вдосконалюється.

3.1.3. C#

C# була розроблена Андерсом Гейлсбергом, Скотом Вілтом та Пітером Гольде. Одне із головних переваг C#, це можливість повторного використання компонентів. Відзначають, що C# - це мова компонентного програмування.

C# була створена паралельно з каркасом FrameworkNet і враховує його можливості, як FCL, CLR.

C# це об'єктно-орієнтовна мова. До переваг мови відносять:

- можливості спадкування й універсалізації є спадкоємцем мови C/C++;
- ефективність коду;
- потужна бібліотека;
- автоматично звільняє пам'ять;
- типи, значення null, забезпечують захист від змінних;
- обробка винятків;
- функціональне програмування, підтримане лямбда виразами;
- загальний шаблон для роботи з даним;
- єдина система типів;

- виділення об'єктів та зберігання спрощена структури;
- підтримка універсальних методів;
- продуктивність;
- синтаксис для створення розподільних систем;
- управління ресурсами;
- гнучкість;

C# - це об'єктно-компонентно орієнтована мова програмування. Вона поєднує універсальні мови програмування і потужність і гнучкість нововведень. Це одна із найкращих мов програмування.

Проаналізувавши мови програмування було обрано мову C#. Яка розвиває принципи C++ і використовує переваги мови Java. Швидкість роботи програми C# найвища з перелічених мов. Немаловажним чинниками є надійного і ефективного використання коду. Потужна бібліотека дає можливість просто одержувати і зберігати інформацію з бази даних.

C# володіє набором засобів, які допомагають у реалізації програми. Вона добре підходить для “Розпізнавання об'єктів на зображенні”.

3.2 Бібліотеки для реалізації

3.2.1. OpenCV

Проектування OpenCV був запущений у 1999 році компанією Intel. На даний момент проект підтримує організація OpenCV.org.

Результат проекту – бібліотека OpenCV (Open Source Computer VisionLibrary). Бібліотека OpenCV – це комп'ютерний зір, це база функцій, алгоритмів обробки зображень з відкритим кодом.

Бібліотека OpenCV включає в себе великий набір, як класичних, так і сучасних алгоритмів комп'ютерного зору та машинного навчання.

В ній зберігається понад 2500 оптимізованих алгоритмів. Бібліотека написана на мові високого рівня C\C++ та має інтерфейси для мов C++, Python, Java, підтримує Windows, Linux, MacOS, iOS та Android.

Бібліотека OpenCV реорганізована замість великих універсальних модулів бібліотека поділена на невеликі модулі:

- `opencv_core` – це ядро, яке містить базові структури, обчислення;
- `opencv_imgproc` – це обробка зображень;
- `opencv_highgui` – використовується для вводу і зберігання зображень і відео;
- `opencv_ml` – методи машинного навчання;
- `opencv_features2d` – дескриптори;
- `opencvvideo` – аналіз руху, відстеження об’єктів;
- `opencv_objdetect` – детектори об’єктів на зображенні;
- `open_calib3d` – калібровка камери та пошук стеріо-відповідності та елементи обробки тривимірних даних;
- `opencv_flann` – це швидкий пошук сусідів;
- `opencv_contrib` – містить код, який не готовий до використання;
- `opencv_legacy` – застарілий код;
- `open-gpu` – прискорення функцій OpenCV за рахунок CUDA (NVidia)

Розглянемо схему роботи з бібліотекою OpenCV. На початку роботи програми, захоплюється зображення – використовують модуль Highgui.

Наступний крок це попередня обробка: нормалізація освітлення позбавлення шуму, вирівнювання, контрасту. Для цього використовують модуль `imgproc`, а потім модуль `features2d` – для виділення особливостей об’єкта.

Наступний крок детектування об’єктів, виділення частин, сегментація до роботи приступають модулі `imgproc`, `objdetect`.

Для виконання задач по знаходженню об’єкта, аналізу його структури у роботу включаються модулі `video`, `calib3d`, `contrib`, `videostab`, `ml`.

Модуль ml розпізнає і приймає рішення чи знайдений об'єкт той, який потрібен рис. 3.1.

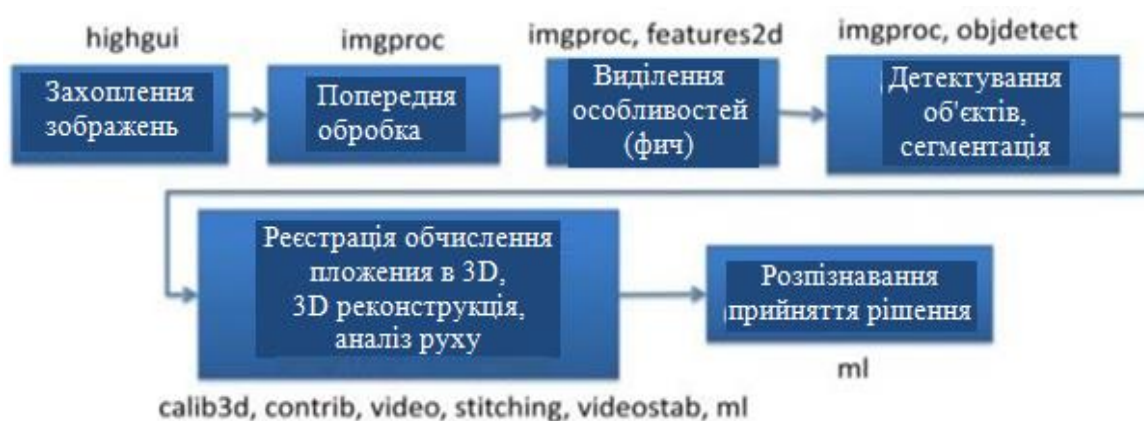


Рисунок 3.1 – Модулі бібліотеки OpenCV

Основна мета OpenCV – це зображення загальної інфраструктури для прискорення використання машинного сприйняття та для програм комп'ютерного зору.

Використовують для, класифікації зображень, відстеження рухомих об'єктів, розпізнавання краєвидів і т д.

В Ізраїлі бібліотеку OpenCV використовують для виявлення вторгнень на територію. В Іспанії та США для виявлення аварій, для моніторингу шахтового обладнання в Китаї, для швидкого виявлення облич в Туреччині та Японії.

Бібліотекою OpenCV користуються понад 56 тисяч користувачів. Бібліотека широко використовується у різноманітних галузях.

3.2.2. TensorFlow

TensorFlow – це бібліотека для швидких числових обчислень, створена і випущена Google. Застосовується, як для досліджень, так і для розробки власних продуктів Google. TensorFlow працює дуже швидко, написаний на мові С.

Основний API для роботи з бібліотекою реалізований для Python. Також існують реалізації для C#, C++, Haskell, Java, Go та Swift. Для досягнення необхідного ступеня абстракції TensorFlow дозволяє використовувати API різного рівня.

Програми TensorFlow запускаються на кластері в хмарі, локальній машині, пристроях iOS та Android процесорах чи графічних процесорах.

Досить довгий час TensorFlow існував в закритому режимі під назвою Dist Belief. В листопаді 2015 року фреймворк був випущений в open source. Фреймворк постійно розвивається і є досить провідною бібліотекою на сьогоднішній день.

TensorFlow є системою машинного навчання Google Brain другого покоління.

Цю бібліотеку, можна використовувати для створення програм глибокого навчання безпосередньо, або за допомогою бібліотек – оболонок, які спрощують процес TensorFlow.

TensorFlow отримав свою назву від операцій з багатовимірними масивами чисел, які також називають “тензорами”. Тензори є “паливом” для графа.

Основою роботи з TensorFlow є побудова і виконання графа обчислень. Граф обчислень описує, як будуть проводитись обчислення. У класичному підході програмування пишеться код, який виконується за рядок. В Tensorflow цей підхід потрібен тільки, як допоміжна мета.

Створення структур, яка задає порядок обчислень – це основа TensorFlow. Програми діляться на дві частини – складання графа обчислень і виконання обчислень в структурах.

В TensorFlow граф складається з плейсхардеров, змінних і операцій. Обчислювальні графи виконуються в сесіях. Існує 2 типи сесії: звичайні та інтерактивні.

Інтерактивна сесія більше підходить для виконання в консолі і визначається, як сесія за замовчуванням. В цьому є різниця між звичайною сесією і інтерактивною. Основний ефект, що об'єкт сесії не передають у функції обчислення, як параметр.

TensorFlow може працювати на багатьох паралельних процесорах, як CPU, та GPU опираючись на архітектури CODA, що робить всі процеси роботи, значно швидше.

TensorFlow використовують в багатьох галузях. Для автоматизованої анотації зображень використовують систему DeepDream TensorFlow. Google для збільшення пошукової видачі Google використовує систему Rank Brain, яка заснована на TensorFlow.

Широко використовують TensorFlow в медицині, так TensorFlow допомагає лікарям сканувати діабетичну ретинопатію, а це допомагає запобігати сліпоті.

Використання TensorFlow в лісовому господарстві, допомагає попередити владу про незаконну вирубку лісу.

TensorFlow використовують для полювання на нові планети. За допомогою TensorFlow була знайдена екзопланета Kepler-90.

TensorFlow використовують для створення нейронних мереж підтримує безліч мов програмування. Але робота з TensorFlow вимагає написання значної кількості програмних кодів. Такий підхід обгрунтований ефективністю, але потрібно це враховувати при зміні архітектури.

TensorFlow – підтримується великою спільнотою розробників і технічними компаніями, а також продовжує розвиватися.

3.2.3. YoLo

Головна особливість YoLo полягає в тому, що більшість систем застосовують CNN кілька разів із різними регіонами зображення. YoLo

використовує її лише один раз для всього зображення. Мережа ділить зображення на сітку і передбачає розташування шуканого об'єкта для кожної ділянки.

Мережа сканує все зображення відразу і враховує контекст при знаходженні і розпізнаванні об'єкта. Це є перевага даного підходу.

Архітектура YoLo складається з вхідного шару, шару згортки, шару об'єднання, зв'язаного шару та вихідного шару (рис.3.2). Шар згортки використовується для виявлення ознак зображення, пов'язний шар використовують для з'ясування положення зображення та значень ймовірності цільових категорій. Шар об'єднання відповідає за зменшення пікселів фрагмента [10].

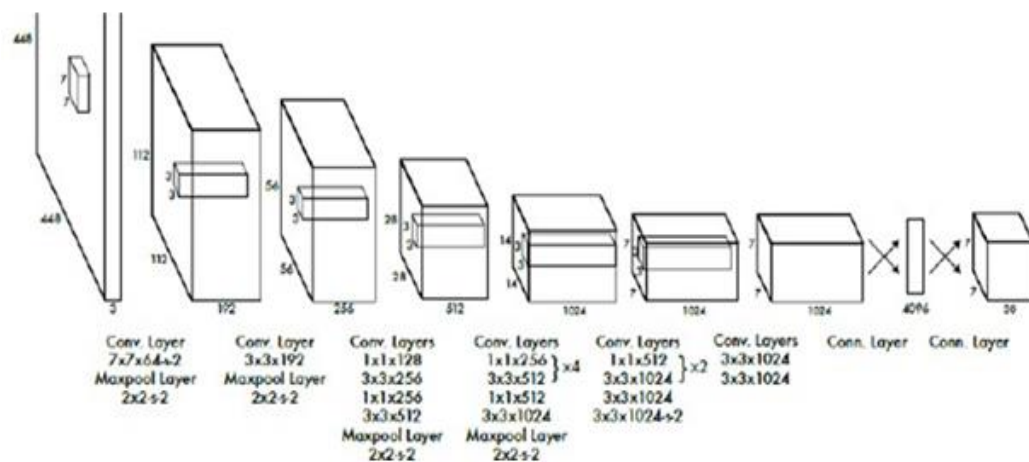


Рисунок 3.2 – Архітектура YoLo

Мережа має 24 згорткових шарів, 2 повністю зв'язних шари. Завдяки чергуванню згорткових шарів 1×1 відбувається зменшення ознак від попередніх шарів. Попередньо тренують згорткові шари для завдання класифікації ImageNet з половиною роздільної здатності, а потім подвоюється роздільна здатність для самого процесу виявлення.

Припустимо, що B – кількість розсувних вікон. C – загальна кількість категорій. Розміри вихідного шару (17):

$$S \cdot S \cdot (B \cdot 5 + C). \quad (17)$$

Вихідна модель виявленої межі (18):

$$T = (x, y, w, h, c), (18)$$

де (x, y) – координати обмежувального боксу; (w, h) – висота і ширина обмежувального вікна виявлення; (c) – оцінка достовірності, яка відображає значення ймовірності поточного вікна (19):

$$C = P_0 \cdot P_{IoU}, (19)$$

P_0 – ймовірність включення об'єкта виявлення; P_{IoU} – співвідношення площі перекриття ковзного вікна і реального об'єкта виявлення (20).

$$P_{IoU} = \frac{Area(BB_i \cap BB_g)}{Area(BB_i \cup BB_g)}, (20)$$

де BB_i – обмежувальний бокс виявленого об'єкта; BB_g – стандартний бокс на основі навчальної вибірки.

YoLo в 1000 раз швидше, ніж R-CNN і в 100 раз швидше ніж Faster-CNN.

Вдосконалена версія YoLo є YoLov3. Вона складається з 106 згорткових шарів. Особливість YoLov3 є те, що на виході є 3 шари, кожен шар розрахований на виявлення об'єктів різного розміру.

Спрощена версія YoLov3 є YoLov3 –tiny на виході має 2 шари. YoLov3–tiny час призначений для невеликих об'ємів даних і гірше приймає дрібні об'єкти. Об'єм мережі займають невеликий обсяг пам'яті, через спрощену будову, вона зменшує час опрацювання кадрів. Тому її краще використовувати в мобільному пристрої.

YoLov3-KD має 3 вихідні шари, орієнтована на меншу кількість об'єктів. Має перевагу у швидкості та розмірі пам'яті.

3.3 Порівняння

Дослідивши бібліотеки OpenCV, TensorFlow та YoLo зрозуміло, що вони підходять для розробки програми розпізнавання об'єктів на зображенні.

Для аналізу роботи бібліотек була розроблена програма на знаходження і розпізнавання об'єкта. Було взято 20 зображень різного забарвлення і розміру для дослідження точності та швидкості розпізнавання.

Була створена таблиця 3.1 для більш наглядного показання порівняння.

Таблиця 3.1 – Порівнянн бібліотек для програми розпізнавання

Програмне забезпечення	OpenCV	TensorFlow	YoLo
Автор	Intel Corporation, Willow Garage Inc., Itseez Ltd.	Google Brain	Joseph Redmon
Відкритий програмний код	Так	Так	Так
Платформа	Windows, Linux, MacOS, iOS та Android	Windows, Linux, MacOS, iOS та Android	Windows, Linux, MacOS, iOS та Android
Підтримка мов програмування	C++, Python, Java	C++, Python, JavaScript, Java, Go, C#, Julia	C++, Python, JavaScript, Java, C#, HTML, CSS, PHP
Згорткові мережі	Так	Так	Так
Активна підтримка спільнотою	Так	Так	Так
Можливість ефективного відлагодження	Так	Так	Так
Низькорівневі оператори	Так	Так	Так

Можливість покращення	Так	Так	Так
Можливість працювати у реальному часі	Так	Ні	Так
Швидкість обробки	Середня	Висока	Найвища
Середня точність	97.6%	97%	98.3%

Найкращі показники швидкості та точності дає YoLov3. Переваги цієї бібліотеки в тому, що YoLov3 аналізує все зображення відразу і вона швидше може обробляти зображення.

В результаті тестування, було зроблено висновок, що найкращі результати правильності та точності оцінки знайденного об'єкта показала програма побудована на основі бібліотеки YoLo.

3.4 Висновок

Програмне забезпечення складнішає з кожним днем. Щоб створити сучасну програму, недостатньо виконувати певні інструкції. Основне це розробити виразну структуру програми, яка буде придатна в виконанні, стійкою до змін і застрахованою від помилок. Після аналізу та порівняння існуючих бібліотек і мов програмування проаналізувавши таблицю 3.1. було зроблено висновок для використання в данній дипломній роботі біло обрано YoLo та мову C#.

РОЗДІЛ 4. АРХІТЕКТУРА РОЗРОБЛЕНОЇ СИСТЕМИ

4.1. Навчання нейронної мережі

Перед тим як почати розробку програмидля розпізнавання об'єктів на зображенні, необхідно встановити з яким набором даних вона буде працювати для розпізнавання. Нейронну мережу потрібно навчати на наборі даних, які були підготовлені спеціалстами тому, що від цього буде залежати точність та ефективність розпізнавання об'єктів.

Підчас розробки був створений алгоритм по якому можна навчати та перевіряти модель рис.4.1.



Рисунок 4.1 – Алгоритм тренування моделі

Для того щоб точність класифікації була високою потрібно підготувати велику кількість фотографій, які будуть сильно відрізнятись одна від одної, фоном, освітленням, об'єктами або об'єктом під різними кутами (ракурсами). Після вибору фотографій вони діляться, як фотографії для тренування та фотографії по яким будуть проводитися тестування.

За допомогою інструменту – LabelImg визначаємо границі знаходження об'єкту, які будуть використовуватись для тренування моделі. Визначивши прямокутні границі, у яких знаходиться об'єкт, слід присвоїти цьому образу назву. Назва цього образу буде використана нейронною мережею для розпізнавання об'єктів. Назва та інформація кожної рамки зберігається у файл формату xml. Такий метод дозволить нейронній мережі працювати більш коректно.

Генерація YoloConvert створює файл з вагою та конфігурації. Ці данні потім будуть використовуватися для розпізнавання об'єктів, класифікація та створюється карта міток, яка буде допомагати у навчанні.

4.2. Архітектура нейронної мережі

Було використано CNN модель, яка має гарні показники точності та швидкості у розпізнаванні об'єктів на зображенні.

Архітектура CNN – це нейронна мережа з шарами згортки. Вона виявляє обмежувальні границі зображення і класифікує їх. Архітектура CNN робить це за один перехід. Завдяки швидкості мережа може обробляти велику кількість зображен. На малюнку розглянуто архітектурну модель рис.4.2.

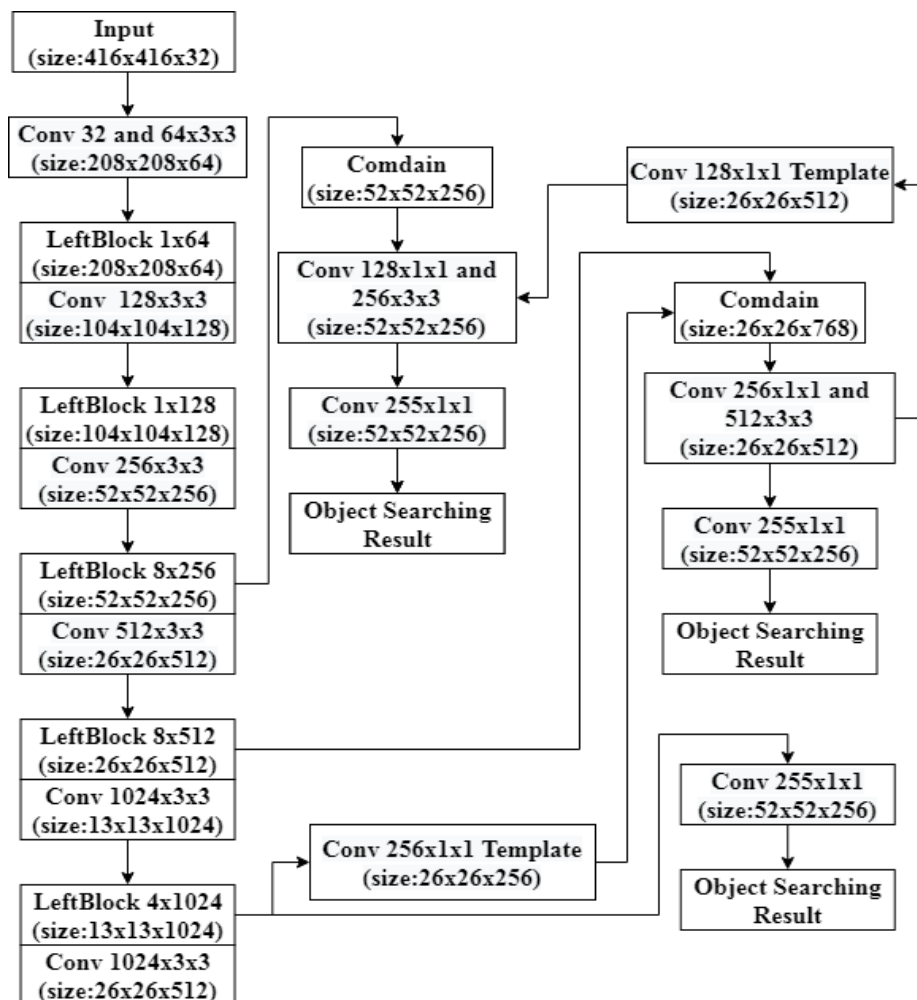


Рисунок 4.2 – Архітектура моделі нейронної мережі

Модель має багато частин. Вихідне зображення подається на згорткові шари, відбувається розширення зображення до розмірів 300×300 або 500×500.

Відбувається обробка вхідного зображення за допомогою фільтрів згортки. Зображення розбивається на комірки, які будуть аналізуватися.

Шари мають свої просторові розмірності. Conv 11,2 має просторову розмірність 1×1. Для цього шару робиться 4 передбачення об'єкта для кожної комірки.

Для Conv 7, Conv 8.2, Conv 9.2 – по 6 передбачень на комірку. Двох інших згорткових шарів – по 4 передбачення на комірку. В результаті отримуємо 8732 передбачення.

На спеціальний модуль детекції та класифікації подаються результати фільтрів згортки, відбувається аналіз для комірки і залишається 1 прогноз.

На модуль швидко стискання поступають передбачення від усіх згорткових шарів і вибирається тільки 1 прогноз, який є результатом.

4.3. Розроблена програма

Підчас запуску програми дається вибір: завантажити зображення або увімкнувши камеру зробити власне фото.

Коли обирається будь який з двох варіантів за допомогою простору Emgu.CV створюється клас VideoCapture(), також створюється масив камери Ds Device[] та клас розпізнання об'єкта Search().

У випадку коли обирається варіант з камерою за допомогою DsDevice.GetDevicesOfCat() програма отримує доступ до камери пристрою. Потім, коли знімок був зроблений VideoCapture_ImageGrabbed() перехоплює зображення після чого воно направляється в Search(), де у ShowSearch() і відбувається сам процес розпізнавання.

Якщо було обрано варіант із завантаженням зображення то за допомогою Emgu.CV створюється клас Mat(). Зображення надходить у програму у вигляді масиву який необхідно обробити. Функція videoCapture.Retrieve() передає вхідні параметри зображення на Mat(), після чого mat.ToImage<Bgr,byte>() та RecognitionResult конвертують масив у зображення та передають його у бокс з іншими зображеннями, звідки його перенаправляють у Search().

У ShowSearch() зображенням обробляють та розпізнають об'єкти. Пошук об'єкта проходить у YoloWrapper(), де зображення обробляє модель нейронної мережі, яку було показано раніше. Після знаходження об'єкт у Graphics() навколо об'єктів створюють рамку та підписують його.

Коли зображення повністю оброблене його направляють до RecodnitionResult де користувач обирає зберегти зображення або ні.

Загальна схема програми зображена на рис.4.3.

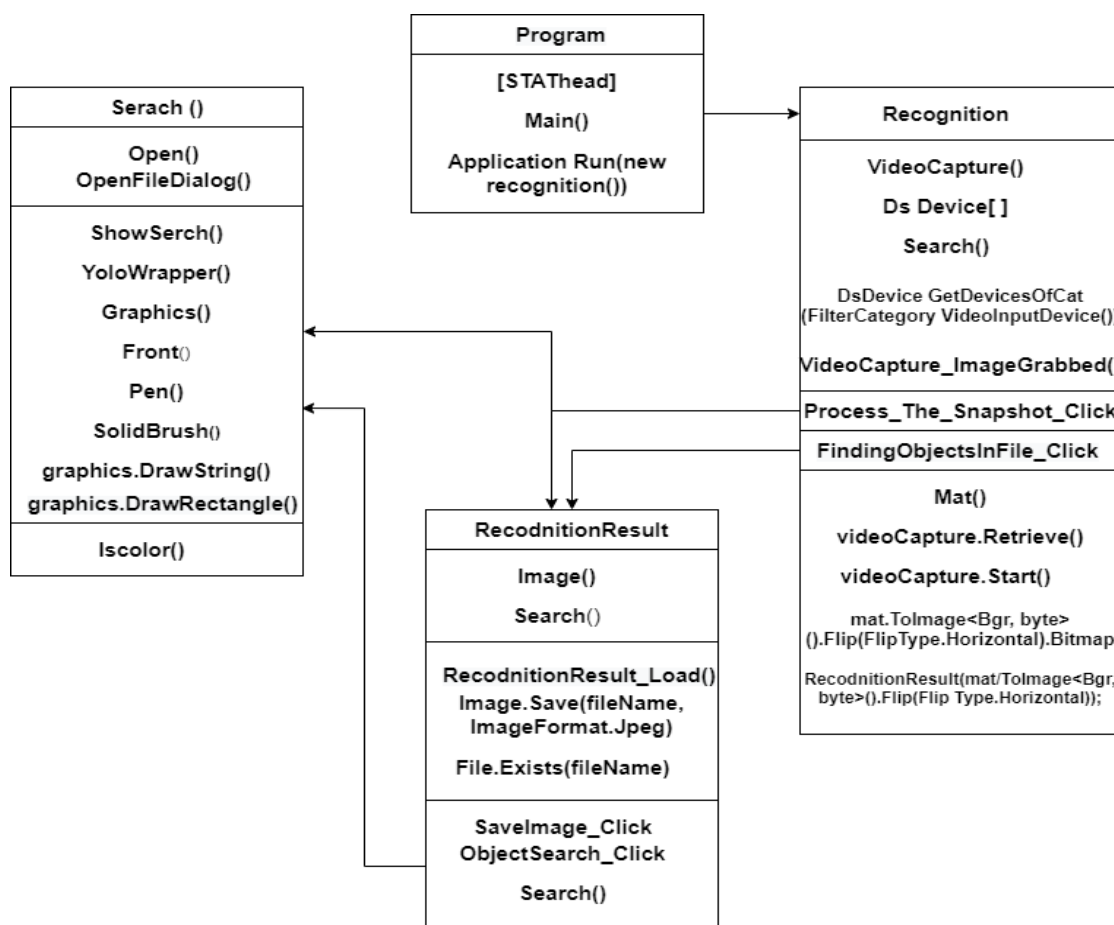


Рисунок 4.3 – Загальна схема програми

4.4. Тестування

В папці “PROGRAM” міститься дві папки “Object Recognition” та “ObjRec”. В “Object Recognition” міститься сама програма, а в “ObjRec” знаходиться вже зібрана програма.

Для запуску програми необхідно перейти до папки “ObjRec” знайти файл Object Recognition.exe та запустити її. З’являється екран розпізнавання об’єктів з простим і зрозуміле діалогове вікно рис.4.4.

Зверху знаходяться 5 кнопок та з правого боку можливість обрати камеру.

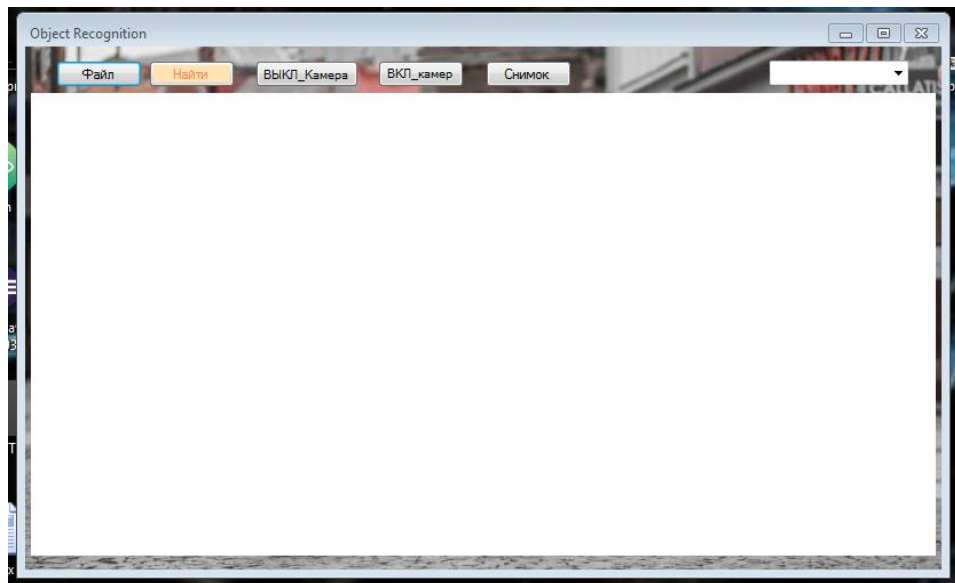


Рисунок 4.4 – Діалогове вікно

Після запуску програми користувачу надається можливість завантажити зображення або зробити знімок самому.

Для того щоб завантажити зображення необхідно натиснути на “Файл” після чого з’явиться можливість знайти місцезнаходження зображення та обрати його рис.4.5.

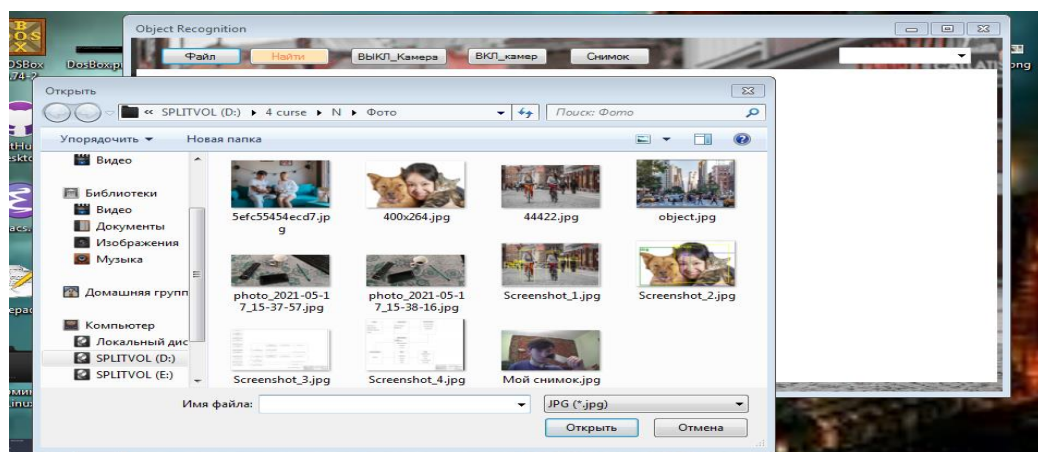


Рисунок 4.5 – Вибір файлу для завантаження в програму

Обравши зображення потрібно натиснути кнопку “Найти” і почекати поки об’єкти на зображенні розпізнаються рис.4.6.

Коли програма знайде всі об’єкти відкриється нове вікно. Об’єкти на зображенні будуть виділені прямокутними рамками різного кольору. У лівому верхньому куті буде написана назва об’єкту, який розпізнала програма.

У відкритому вікні із обробленим зображенням користувачу запропонується зберегти зображення з виділеними об’єктами або просто зачинити нове вікно рис.4.7.

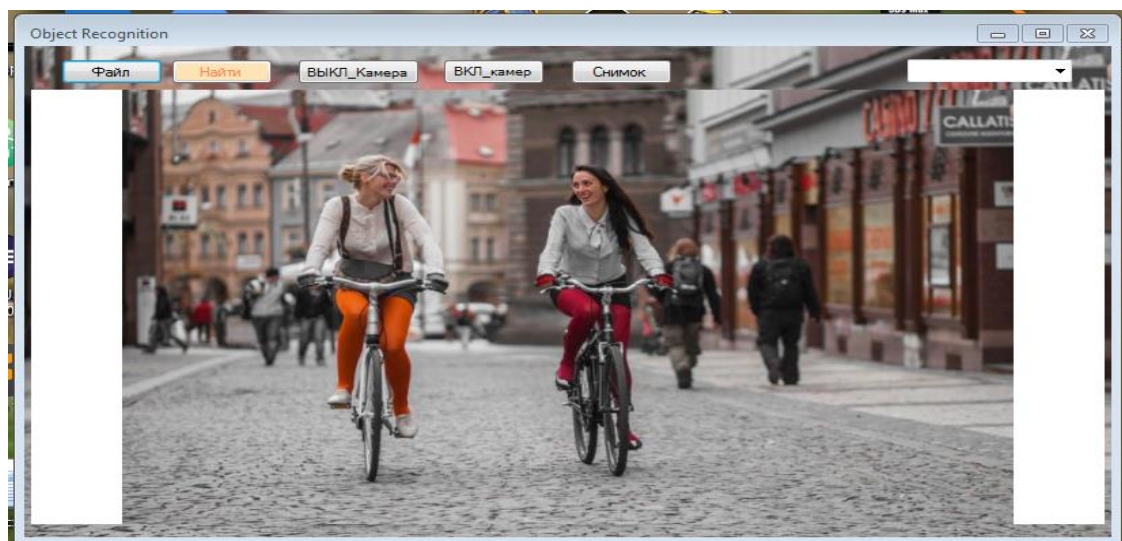


Рисунок 4.6 – Зображення до розпізнавання

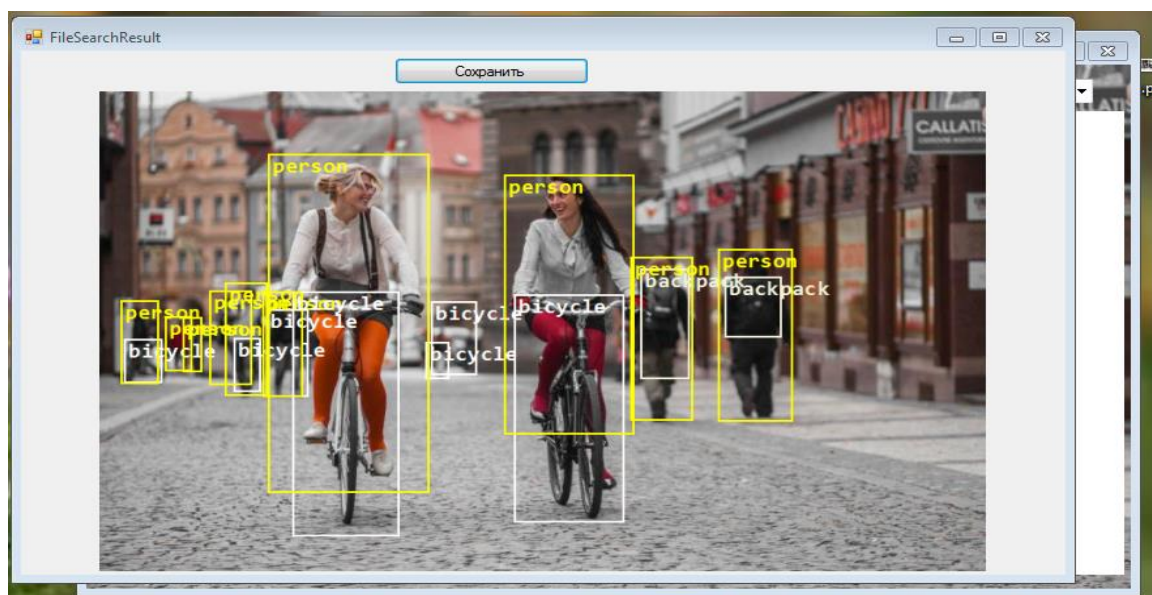


Рисунок 4.7 – Зображення з виділеними об’єктами

Після натискання “Сохранить” або закриття користувач повернеться до першого інтерфейсу із зображенням, яке щойно було оброблене. Користувач може знову обрати нове зображення або увімкнувши камеру зробити знімок та розпізнати об’єкти на ньому.

Для того щоб увімкнути камеру необхідно спочатку обрати яку саме камеру програма має використовувати.

Зараз кожний ноутбуки або екрани компютерів мають вже вбудовану у себе камеру, проте камера може вийти з ладу, давати погану якість зображення або просто зображення не зможе потрапити в об’єктив камери через певні обставини.

Вибір камери можна здійснити натиснувши на біле пусте поле з правого верхнього боку і в з’явившомуся списку обрати потрібну вам камеру, що і показано на рис.4.8.

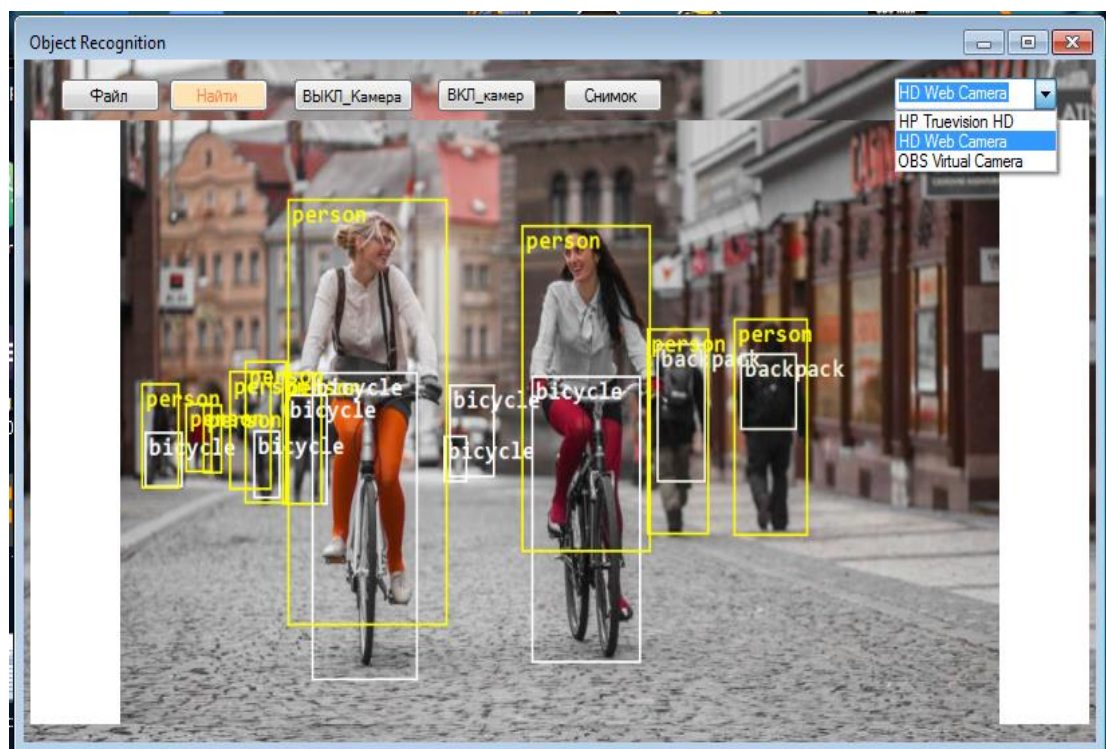


Рис.4.8 – Обрання камери

Після того як користувач обрав камеру слід її увімкнути натиснувши “ВКЛ_камера”. Після натискання на екрані ви побачите те що показуватиме об’єктив камери. Натиснувши кнопку “Снімок” програма автоматично зробе знімок та завантажить її до програми рис.4.9.

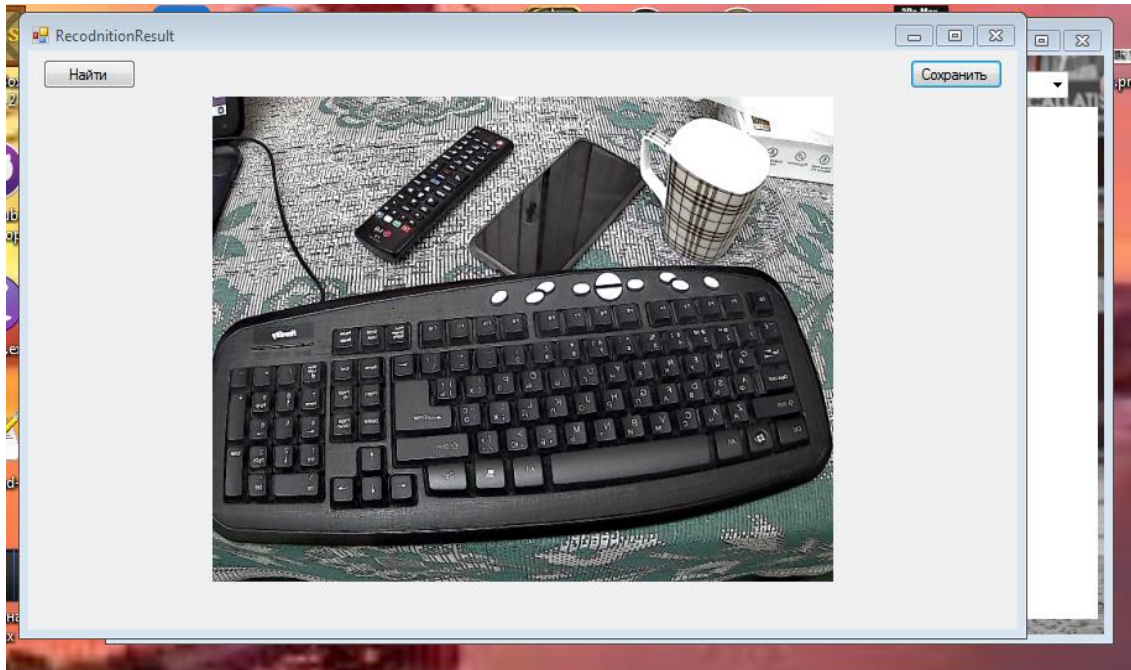


Рисунок 4.9 – Знімок за допомогою веб камери

Коли знімок зроблено, користувач може просто зробити новий знімок натиснувши “Сохранить” або закрити натиснувши вікно з фото. Для того щоб обробити щойно зроблене фото необхідно натиснути “Найти” та почекати, після чого програма виділить об’єкти на фотографії рис.4.7.

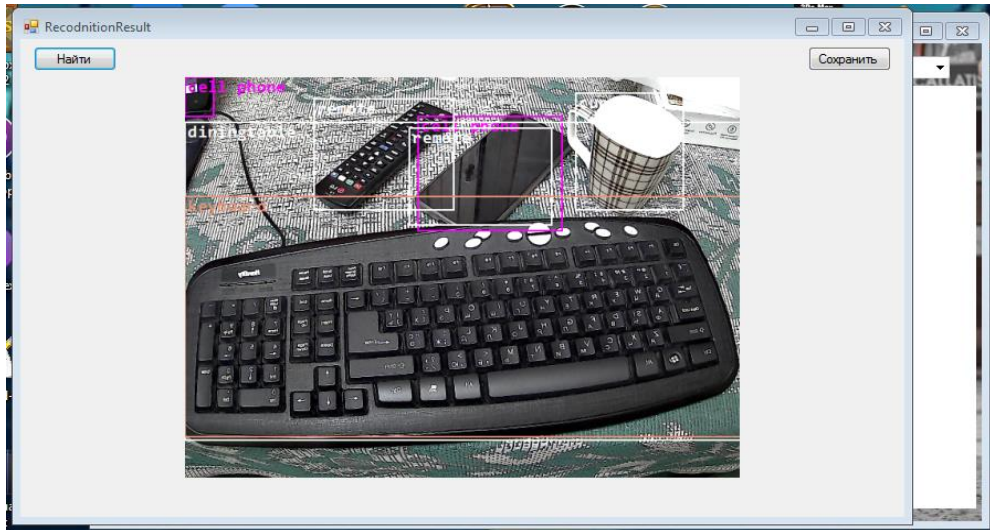


Рисунок 4.9- Розпізнавання об'єктів на фото

Користувач може зберегти оброблене фото або закрити його та зробити нове.

Після того як користувач використав програму він може легко з неї вийти закривши її вікно.

Фотографії, які були зроблені можна знайти разом із обробленими зображеннями, які були збережені у папці "ObjRec".

Після декількох завантажень зображень, зроблених фотографій та їх обробки можна перекоонатися що програма працює так, як і було заплановано рис.4.10.

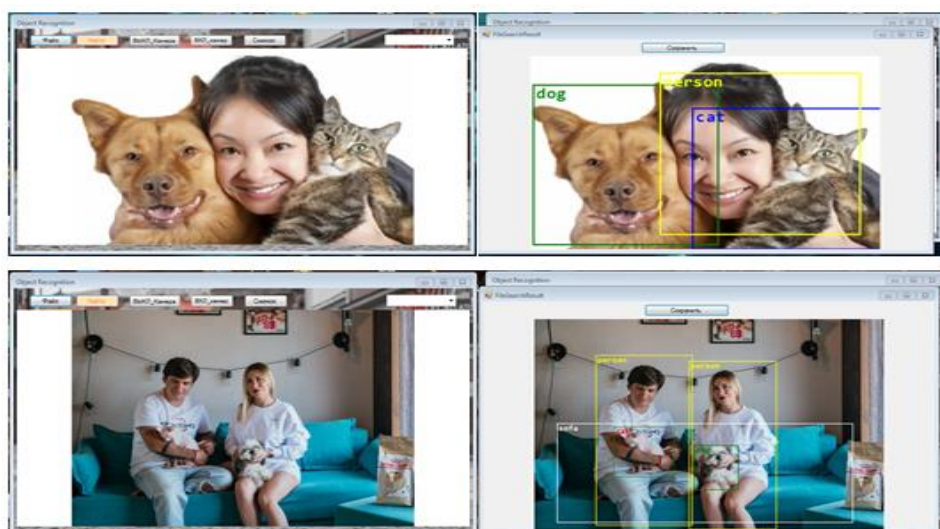


Рисунок 4.10 – Тестування програми

Програма, яка була створена у ході роботи є швидкою, точною та досить простою у використанні.

4.5. Висновок

В результаті була створена програма розпізнавання об'єктів на зображенні. Для створення програми використовувалась бібліотека YOLOv3. Програма написана на мові C#. Використання згорткової мережі CNN дало можливість з великою точністю здійснювати розпізнавання об'єктів на зображенні.

Тестування програми виявились позитивними. Дану програму можна використовувати в багатьох сферах.

ВИСНОВКИ

Метою даної дипломної роботи було створення програми “Розпізнавання об’єктів на зображенні” за допомогою комп’ютерного зору.

Для здійснення цієї задачі була розглянута актуальність даної теми. Був проведений аналіз предметної області і розглянуті існуючі програмні рішення.

Було досліджено методи і алгоритми обробки зображень. Розглянута проблематика та методи розпізнавання об’єктів на зображенні.

Для реалізації програми була здійснена обробка розглянутих мов та засобів реалізації. Проаналізовано ефективність використання бібліотек для розробки програми “розпізнавання об’єктів на зображенні”.

Для розробки програми було обрано бібліотеку YoLov3 та мову програмування C#. В результаті була розроблена архітектура програми та алгоритм роботи.

Тестування програми показали гарні результати. Програма надійна і проста у використанні.

Дана програма є універсальною і може бути використана в різних областях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кутковецький В. Я. Розпізнавання образів. Навчальний посібник / 2017 – 420 с.
2. Вовк С.М., Гнатушенко В.В., Бондаренко М.В. Методи обробки зображень та комп'ютерний зір/ 2016 – 150 с.
3. Оптический поток [Електронний ресурс]. – Режим доступу: <http://habrahabr.ru/post/201406> – Дата доступу: травень 2021.
4. A Gentle Introduction to Object Recognition With Deep Learning – [Електронний ресурс]. – Режим доступу: <https://machinelearningmastery.com/object-recognition-with-deep-learning/> – Дата доступу: травень 2021.
5. Гавриленко В.В. Іванченко Г.Ф. Шевченко Г.Є. Теорія розпізнавання образів. Навчальний посібник– [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5607072/>– Дата доступу: травень 2021.
6. Теорія розпізнавання образів – [Електронний ресурс]. – Режим доступу: https://uk.m.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F_%D1%80%D0%BE%D0%B7%D0%BF%D1%96%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D1%96%D0%B2 – Дата доступу: травень 2021.
7. Методи обробки зображень – [Електронний ресурс]. – Режим доступу: https://nmetau.edu.ua/file/07_7.5_lbr_gr_rbr_.pdf – Дата доступу: травень 2021.
8. Выделение объектов на изображении по методу Виолы-Джонса (Object selection in the image according to the Viola-Jones method) – [Електронний ресурс]. – Режим доступу: <https://api-2d3d-cad.com/viola-jones-method/>– Дата доступу: травень 2021.

9. Методы нахождения особых точек изображения и их дескрипторов – [Электронный ресурс]. – Режим доступа: <https://nauchkor.ru/pubs/metody-nahozhdeniya-osobyh-tochek-izobrazheniya-i-ih-deskriptorov-587d36515f1be77c40d58c79> – Дата доступа: травень 2021.
10. Обнаружение объектов YOLO с помощью OpenCV и Python дескрипторов – [Электронный ресурс]. – Режим доступа: <https://www.machinelearningmastery.ru/yolo-object-detection-with-opencv-and-python-21e50ac599e9/> – Дата доступа: травень 2021.