

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ПРИЛАДОБУДІВНИЙ ФАКУЛЬТЕТ

КАФЕДРА АВТОМАТИЗАЦІЇ ТА СИСТЕМ НЕРУЙНІВНОГО КОНТРОЛЮ

«До захисту допущено»

Завідувач кафедри

_____ Юрій КИРИЧУК

« ____ » _____ 2023 р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра за освітньо-професійною програмою "Комп'ютерно-інтегровані технології проектування приладів" спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» на тему: «Автоматизована фотометрична система для розпізнавання об'єктів»

Виконав: студент IV курсу, групи ПМ-91 Гапонов Дмитро Олександрович _____

Керівник: доцент к.т.н., доцент Литвиненко Павло Леонідович. _____

Рецензент: д.т.н., проф. Володарський Євген Тимофійович _____

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

Студент _____

(підпис)

Київ – 2023 року

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4	ДПБ.ПМ-91.05.1760.01.000.ПЗ	Пояснювальна записка	73	
2	A1	ДПБ.ПМ-91.05.1760.02.000. СхС	Схема структурна	1	
3	A1	ДПБ.ПМ-91.05.1760.03.000. СхА	Алгоритм роботи системи	1	
4	A1	ДПБ.ПМ-91.05.1760.04.000. ГрВ	Графіки «втрат»	1	
5	A1	ДПБ.ПМ-91.05.1760.05.000. ГрМ	Графіки «метрик»	1	
6	A1	ДПБ.ПМ-91.05.1760.06.000. СхІ	Схема ілюстративна CNN	1	

				ДПБ.ПМ-91.05.1760.01.000.ПЗ	
	ПІБ	Підп.	Дата		
Розробник	Гапонов Д. О.			Відомість дипломної роботи	Лист
Керівник	Литвиненко П.Л.				Листів
Консульт					1
Н/контр.					1
Зав. каф.	Киричук Ю.В.				КПІ ім. Ігоря Сікорського Каф. АСНК Гр. ПМ-91

Пояснювальна записка
до дипломної роботи
на тему: «Автоматизована фотометрична система
для розпізнавання об'єктів»

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Приладобудівний факультет
Кафедра автоматизації та систем неруйнівного контролю

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютерно-інтегровані технології проектування приладів»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Юрій КИРИЧУК

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Гапонову Дмитру Олександровичу

1. Тема роботи «Автоматизована фотометрична система для розпізнавання об'єктів», керівник роботи Литвиненко Павло Леонідович, кандидат технічних наук, доцент, затверджені наказом по університету від «___» _____ 20__ р. № _____

2. Термін подання студентом роботи _____.

3. Вихідні дані до роботи: Автоматизована фотометрична система, з можливістю виводу зображення через ПК. Система містить мінімум 4 камери (з можливістю живлення через РОЕ), режими роботи «день»/«ніч». Мінімальна освітленість «день» - 0,01лк / «ніч» - 0,025лк. Живлення камери до 15,4В. Комутатор мінімум з 4 входами типу РОЕ af (РОЕ)/at(РОЕ+). Нейронна мережа для аналізу фотометричної системи. Функція втрат нейронної мережі розташування об'єктів має значення коефіцієнту максимум – 0,7. Точність нейронної мережі (Precision) має значення мінімум 0,9(90%).

4. Зміст пояснювальної записки

Вступ. Перелік скорочень. Аналітичний огляд. Практична частина. Висновки. Список використаних джерел.

Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) 5 плакатів:

Схема структурна А1. Алгоритм роботи системи А1. Графіки «втрат» А1, Графіки «метрик» А1. Схема ілюстративна CNN А1.

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Формулювання завдання роботи	До 12.02.2023	Виконано
2	Проведення аналітичного огляду	До 18.05.2023	Виконано
3	Підбір компонентів	До 21.05.2023	Виконано
4	Проведення математичного моделювання нейронної мережі	До 28.05.2023	Виконано
5	Побудова графіків до графічної частини	До 01.06.2023	Виконано
6	Оформлення ПЗ	До 6.06.2023	Виконано
7	Оформлення графічної частини	До 8.06.2023	Виконано

Студент

Дмитро ГАПОНОВ

Керівник

Павло ЛИТВИНЕНКО

АНОТАЦІЯ

На сьогоднішній день досягнення високої точності у розпізнаванні об'єктів у реальному часі є актуальною проблемою в багатьох галузях, таких як комп'ютерне зорове сприйняття, автономні транспортні засоби, медична діагностика і багато інших. Швидкий розвиток технологій та збільшення об'єму доступних даних зробили можливим використання машинного навчання та нейронних мереж для досягнення цієї мети.

В даній дипломній роботі проведено розробку автоматизованої фотометричної системи для розпізнавання об'єктів. У першій частині роботи розглянуто різні типи автоматизованих системи для розпізнавання об'єктів та проведено їх аналіз. Проаналізовані різні підходи до реалізації комп'ютерного зору. Детально розглянуто нейронні мережі, включаючи різні види нейронних мереж та згорткові мережі. Окрема увага була приділена огляду фреймворків, що використовуються в даній області, а також застосунків для збору даних та їх розмітки.

У практичній частині роботи була розроблена структурна схема системи, алгоритм її роботи, проведений вибір технічних та програмних компонентів. Також був проведений огляд процесу збору даних. Було здійснено математичне моделювання навчання нейронної мережі та проведено аналіз отриманих результатів.

ABSTRACT

Today, achieving high accuracy in real-time object recognition is an urgent problem in many fields, such as computer vision, autonomous vehicles, medical diagnostics, and many others. The rapid development of technology and the increasing amount of available data have made it possible to use machine learning and neural networks to achieve this goal.

In this thesis, we have developed an automated photometric system for object recognition. In the first part of the work, different types of automated systems for object recognition are considered and analyzed. Different approaches to the implementation of computer vision are analyzed. Neural networks, including various types of neural networks and convolutional networks, are discussed in detail. Particular attention was paid to the overview of frameworks used in this area, as well as applications for data collection and data markup.

In the practical part of the work, the system's block diagram and algorithm were developed, and the selection of hardware and software components was made. The data collection process was also reviewed. Mathematical modeling of neural network training was performed and the results were analyzed.

ПЕРЕЛІК СКОРОЧЕНЬ

CNN - Згорткова нейронна мережа (Convolutional Neural Network) - це тип нейронної мережі, яка спеціалізується на обробці даних з використанням згорток для виявлення візуальних шаблонів у вхідних зображеннях або даних з регулярною структурою.

RNN - Рекурентна нейронна мережа (Recurrent Neural Network) - це тип нейронної мережі, яка використовує зворотний зв'язок, щоб передавати інформацію з попередніх кроків часу до поточного кроку, дозволяючи моделі працювати з послідовними даними, такими як текст або часові ряди.

PTZ - Панорамування, нахил і збільшення (Pan-Tilt-Zoom) - це функціональні можливості камери, які дозволяють панорамувати (рухати в горизонтальній площині), нахил (рухати в вертикальній площині) і збільшувати зображення.

PAL - Фазова альтернативна лінія (Phase Alternating Line) - це стандарт телевізійного відео, який використовується в багатьох країнах, включаючи більшість країн Європи, для передачі кольорового телевізійного сигналу.

NTSC - Національний телевізійний системний комітет (National Television System Committee) - це стандарт телевізійного відео, що використовується в США та деяких інших країнах, для передачі телевізійного сигналу

BNC - Вайонет-Неймовірний (Bayonet Neill-Concelman) - це тип роз'єму, що використовується для з'єднання коаксіального кабелю, часто використовується в системах відеосигналу.

RCA - роз'єм RCA - це тип роз'єму, що використовується для передачі аналогових аудіо- та відеосигналів.

VGA - Video Graphics Array (VGA) - це стандарт інтерфейсу для передачі відеосигналу від комп'ютера до монітора або відеопроєктора.

YOLO - You Only Look Once - це алгоритм об'єктного визначення, який використовується для швидкого та точного виявлення об'єктів у зображеннях та відео.

OpenCV - Відкрите комп'ютерне зорове (Open Computer Vision) - це бібліотека програмних інструментів для обробки зображень та комп'ютерного зору, яка надає набір функцій для роботи з відео, зображеннями, обробки зображень, розпізнавання об'єктів тощо.

CVAT - Computer Vision Annotation Tool - це програмний інструмент, який допомагає анотувати зображення та відео для завдань комп'ютерного зору, таких як розпізнавання об'єктів чи семантична сегментація.

IDEA - Інтегроване середовище розробки (Integrated Development Environment), це програмне забезпечення, яке надає зручний інтерфейс та інструменти для розробки, налагодження та тестування програмного коду.

GPU - Графічний процесор (Graphics Processing Unit) - це пристрій, який спеціалізується на обробці графічних даних та виконанні паралельних обчислень, і використовується для прискорення обчислень у галузі машинного навчання та інших областях.

AI - Штучний інтелект (Artificial Intelligence) - це галузь науки та технологій, яка займається розробкою систем та програм, здатних виконувати завдання, які зазвичай вимагають інтелекту людини.

DL - Глибоке навчання (Deep Learning) - це підгалузь машинного навчання, яка використовує нейронні мережі з великою кількістю шарів для автоматичного винайдення репрезентацій даних та вирішення складних завдань.

ЗМІСТ

ВСТУП	11
1 АНАЛІТИЧНА ЧАСТИНА	12
1.1 Типи систем, огляд підходів	12
1.2 Нейронні мережі та їх види	20
1.3 Огляд фреймворків	34
1.4 Огляд застосунків для збору даних та розмітки	39
Висновки до розділу 1	43
2. ПРАКТИЧНА ЧАСТИНА	44
2.1 Структурна схема та алгоритм роботи системи	44
2.2 Вибір технічних компонентів	47
2.3 Програмні компоненти	51
2.4 Математичне моделювання навчання нейронної мережі, огляд отриманих результатів	56
Висновки до розділу 2	68
ВИСНОВКИ	69
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ	70

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>							
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>								
<i>Розроб.</i>		<i>Гапонов Д. О.</i>			АФСРО Пояснювальна записка							
<i>Перевірів.</i>		<i>Литвиненко П.Л.</i>				<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><i>Літ.</i></td> <td style="text-align: center;"><i>Арк.</i></td> <td style="text-align: center;"><i>Аркушіів</i></td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;"> </td> <td style="text-align: center;"> </td> </tr> </table>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушіів</i>			
<i>Літ.</i>	<i>Арк.</i>	<i>Аркушіів</i>										
<i>Н. контр.</i>					НТУУ «КПІ ім. Ігоря Сікорського»							
<i>Затв.</i>												

ВСТУП

На сьогоднішній день досягнення високої точності у розпізнаванні об'єктів у реальному часі є актуальною проблемою в багатьох галузях, таких як комп'ютерне зорове сприйняття, автономні транспортні засоби, медична діагностика і багато інших. Швидкий розвиток технологій та збільшення об'єму доступних даних зробили можливим використання машинного навчання та нейронних мереж для досягнення цієї мети.

Одним з найпоширеніших підходів до розпізнавання об'єктів у реальному часі є використання методів об'єктного виявлення. Об'єктне виявлення - це процес визначення та локалізації об'єктів на зображенні або відео. Один з найуспішніших алгоритмів об'єктного виявлення, який отримав значну популярність, - це алгоритм YOLO (You Only Look Once).

YOLO є нейронною мережею, яка одночасно виконує виявлення та класифікацію об'єктів. Цей підхід дозволяє здійснювати розпізнавання об'єктів у реальному часі з високою швидкістю. Алгоритм YOLO використовує згорткову нейронну мережу для отримання розміщення та класифікації об'єктів на зображенні.

Мета роботи полягає у розробці фотометричної системи з використанням машинного навчання та візуалізації даних для розпізнавання об'єктів у реальному часі, зокрема з використанням алгоритму YOLO. Розробка системи передбачає впровадження алгоритму YOLO до фотометричної системи.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІТИЧНА ЧАСТИНА

1.1 Типи систем, огляд підходів

Прототипи систем для автоматичної фотометричної системи розпізнавання об'єктів можна класифікувати за їх функціональністю та підходами, що використовуються для розпізнавання об'єктів. Вона включає наступні класи прототипів.

Тип з використанням глибокого навчання

Прототип на основі глибокого навчання: використовує глибокі нейронні мережі для розпізнавання об'єктів. Він може включати в себе згорткові нейронні мережі (CNN), які навчаються на великих наборах зображень для автоматичного виявлення та класифікації об'єктів. Прототипи на основі глибокого навчання зазвичай вимагають значних обчислювальних ресурсів та тривалого часу для тренування, але вони можуть досягати високої точності та надійності у розпізнаванні об'єктів.

Прототип на основі глибокого навчання має наступні особливості створення:

1. Архітектура нейронної мережі. Використовується спеціально розроблена архітектура глибокої нейронної мережі, така як Convolutional Neural Network (CNN) або інші подібні архітектури, які ефективно працюють з зображеннями.
2. Навчання на маркованих даних. Модель нейронної мережі навчається на великому наборі маркованих зображень, де кожному об'єкту призначено певну мітку або клас.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Автоматичне визначення ознак. Нейронна мережа самостійно визначає набір ознак, які найкраще розрізняють об'єкти на зображеннях, шляхом підбору вагів нейронної мережі.
4. Аугментація (augmentation) даних. Для поліпшення загальної роботи системи використовується аугментація даних, яка включає розширення набору даних шляхом генерації додаткових зображень зі зміненими параметрами, такими як повороти, масштабування, зміщення тощо.

Прототипи на основі глибокого навчання забезпечують кращі результати порівняно з базовими системами, оскільки він використовує складніші алгоритми, здатні розпізнавати об'єкти з більшою точністю та надійністю. Крім того, прототипи на основі глибокого навчання легко модифікувати та розширяти данні для розпізнавання нових класів об'єктів або вирішення інших завдань споріднених з обробкою зображень.

Системи на основі нейронних мереж можуть бути різних типів та використовуватися через ПК (персональний комп'ютер) або вбудовані в інші пристрої. Основні типи систем на основі глибокого навчання включають наступні:

- автоматизована фотометрична система з використанням комп'ютерної або серверної частини;
- автоматизована фотометрична система з використанням вбудовані системи окремий електронно-обчислюваний прилад;

Автоматизована фотометрична система з використанням комп'ютерної або серверної частини.

Комп'ютерні системи з використанням глибокого навчання. Ці системи використовуються на ПК або серверах і мають велику обчислювальну потужність для навчання та використання нейронних мереж. Вони зазвичай використовуються для складних завдань розпізнавання об'єктів, обробки

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	<i>Арк</i>
						13
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

зображень, машинного навчання та інших задач, які потребують великої обчислювальної потужності.

Прототипом такої системи є Hikvision. Hikvision (рис. 1.1) є одним з провідних виробників автоматизованого обладнання. Вони пропонують різні моделі камер з різними характеристиками та програмне забезпечення, яке включає відео аналітику, систему розпізнавання облич тощо.[1]



Рис. - 1.1 Hikvision: Hikvision [1]

Опис складової системи містить: [1]

- 1 x HD-рекордер DS-7604NI-K1/4P HD Plug & Play NVR;
- 4 x 8-мегапксельні DS-2CD2186G2-I 2.8 мм мережеві купольні камери для внутрішнього та зовнішнього використання;
- 1 x WD Purple 1TB SATA HDD;
- 1 x програмне забезпечення для смартфона;
- 1 x програмне забезпечення для ПК
- підтримка веб-сайту для браузерів Internet Explorer, Safari, Firefox;

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

- 4 x 10-метрові кабелі підключення камер CAT5;
- 1 x LAN-кабель мережі;
- 1 x USB-миша;
- HDMI-кабель для підключення монітора, 1,5 метра.
- Ціна: € 1.375,0.

Автоматизована фотометрична система з використанням окремого вбудованого електронно-обчислюваного приладу.

Вбудовані системи з використанням глибокого навчання. Ці системи включають в себе вбудовані пристрої, які мають обмежену обчислювальну потужність і ресурси. Вони зазвичай використовуються для розпізнавання об'єктів у вбудованих системах, таких як мобільні пристрої, системи безпеки, роботи, автомобільної техніки тощо. Вбудовані системи можуть мати обмежені ресурси, але вони ефективно використовують нейронні мережі для розпізнавання об'єктів у реальному часі.

Прототипом такої системи є Dahua 2MP (рис.1.2): зовнішня система відеоспостереження з 4 камерами типу Turret з мікрофоном і нічним баченням до 30 метрів. [2]

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						15
Зм.	Арк.	№ докум.	Підпис	Дата		



Рис. 1.2 - Dahua 2MP [2]

Специфікації камери Dahua HAC-HDW1231TLMQ-A-0280B [2]

- Сенсор зображення: 1/2.8" 2MP CMOS
- Роздільна здатність: 1920 (Ш) × 1080 (В)
- 1 x BNC (HDCVI): 1 x BNC (HDCVI)
- Мінімальна освітленість: 0.002 Lux/F1.6 30 IRE, 0 Lux з ІЧ-підсвіткою
- ІЧ-підсвічування/Нічний режим: LED ІЧ-масив 30 метрів
- Основні функції: OSD-меню, День/Ніч з ІЧ-фільтром (ICR)
- Налаштовувані функції OSD: Інтелектуальний ІЧ, AWB, AGC, BLC, HLC, 2DNR
- Аудіо: вбудований мікрофон

						Арк
					ДПБ.ПМ-91.05.1760.01.000.ПЗ	16
Зм.	Арк.	№ докум.	Підпис	Дата		

- Об'єктив: 2.8 мм / Н: 107°; V: 56°; D: 127°
- Живлення: DC 12 V
- Стійкість до погодних умов: IP67 / Металевий корпус
- Робоча температура: -40° C ~ + 60° C
- Розміри: 109.9 мм × 88.1 мм
- Вага: 280 г

Dahua Wizensense XVR5104HS-I2 4-канальний Smart DVR

- Головний процесор: Вбудований процесор, Embedded LINUX
- Класифікація об'єктів: детекція людей/транспорт для дротового периметру та вторгнення
- AI-пошук: Пошук за класифікацією цілей (людина, транспорт), обробка до 8 фотографій обличчя/сек
- Режим: Виявлення обличчя незнайомих
- AI-пошук: До 8 зображень цільових облич
- Активація подій: Дзвінок, голосові повідомлення, електронна пошта, знімок, запис, тривога,
- Активація PTZ тощо.
- AI-пошук: Пошук за класифікацією цілей (людина, транспорт)
- Аналоговий вхід для камери: 4 канали, BNC
- HDCVI-вхід для камери: 5 Мп, 4 Мп, 1080P при 25/30 кадрів/с, 720P при 50/60 кадрів/с, 720P при 25/30 кадрів/с
- CVBS-вхід для камери: PAL/NTSC

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

- IP-вхід для камери: 4+2 канали, кожен канал до 6 Мп
- Двосторонній зв'язок: Відновлення аудіо/вихід, RCA
- Компресія: H.265+/H.265/H.264+/H.264
- Бітрейт: 32 Кбіт/с – 6144 Кбіт/с на канал
- З'єднання: 1 HDMI, 1 VGA
- З'єднання: 1 RJ-45-порт (100 Мбіт)
- Смартфон: Додаток для iPhone, iPad, Android
- Сумісність: ONVIF 16.12, сумісні з CGI
- Активація подій: Запис, PTZ, Тур, Відео-передача, Електронна пошта, FTP, Знімок, Дзвінок та Підказки на екрані
- Відео аналітика: Виявлення руху, МД-зони: 396 (22 × 18), Відео втрата, Саботаж та Діагностика
- Режим резервного копіювання: USB-пристрій/Мережа
- Внутрішній жорсткий диск: 1 SATA-порт, ємність до 6 ТБ
- USB: 2 порти USB (USB 2.0)
- Живлення: 12 В/1,5 А
- Розміри: 260 мм × 238 мм × 48 мм
- Ціна € 741,00

Базовий тип

Базовий тип. Цей прототип є початковою версією системи і має базовий набір функцій для розпізнавання об'єктів. Він може використовувати прості алгоритми для виявлення та класифікації об'єктів, такі як методи порогової

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

обробки, геометричні методи тощо. Базовий прототип може бути використаний для оцінки ефективності системи та визначення основних вимог до неї. Він базується на використанні порогового значення яскравості пікселів зображення для виділення об'єктів.

Алгоритм може мати наступні кроки:

1. Перетворення в градації сірого: зображення перетворюється в градації сірого, щоб працювати з одним каналом інформації.
2. Вибір порогового значення: встановлюється певне порогове значення яскравості, яке відрізняє об'єкти від фону.
3. Бінаризація зображення: кожен піксель зображення порівнюється з пороговим значенням. Якщо значення пікселя перевищує поріг, він вважається об'єктом, в іншому випадку - фоном.
4. Виявлення контурів об'єктів: виділені об'єкти проходять процес виявлення контурів, щоб відрізнити їх від інших областей.
5. Класифікація об'єктів: знайдені контури обробляються для визначення типу об'єктів або для подальшого аналізу.

Цей прототип системи є простим, але може бути ефективним для деяких простих сценаріїв, де об'єкти мають достатню відмінність яскравості від фону. Однак, він може мати обмежену точність та надійність у випадках складних завдань або коли об'єкти мають різну текстуру та форму. Тому, для більш складних задач розпізнавання об'єктів бажано використовувати більш продуктивні та потужні алгоритми, такі як нейронні мережі.

Обидва типи розглянутих систем можуть використовувати нейронні мережі, але різні аспекти, такі як обчислювальна потужність, розмір, ресурси та додаткові обмеження, можуть впливати на вибір і використання конкретної системи. Вибір автоматизованої фотометричної системи з використанням комп'ютерної або серверної частини має декілька переваг:

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

- Автоматизація. Комп'ютерні системи з нейронними мережами забезпечують високий рівень автоматизації у процесі розпізнавання об'єктів. Їх можна налаштувати для автоматичного виконання різних складних стандартних та нестандартних завдань, які не передбачені розробником. Це дозволяє забезпечити ефективну та незалежну роботу системи, знизити необхідність у ручній обробці та контролі.
- Гнучкість. Комп'ютерні системи з нейронними мережами надають широкі можливості для налаштування та розширення. За допомогою програмного забезпечення та обладнання їх можна адаптувати до різних завдань розпізнавання об'єктів, використовуючи різні типи нейронних мереж та алгоритмів. Такі системи мають змогу оновлюватися та покращуватися з часом, щоб враховувати нові дані, алгоритми та технологічні вдосконалення. Гнучкість цих систем дозволяє їм адаптуватися до змінних потреб та вимог у розпізнаванні об'єктів.

1.2 Нейронні мережі та їх види.

Нейронні мережі

Нейронні мережі є математичними моделями, що імітують роботу людського мозку, копіюючи роботу людських мультиполярних нейронів, що зображено на рис. 1.3 [3] Вони складаються зі штучних нейронів на рис. 1.4 [4], які з'єднані між собою і формують взаємозв'язану структуру. Кожен нейрон приймає вхідні сигнали, обчислює їх вагу та застосовує активаційну функцію, яка визначає, чи передає нейрон сигнал наступним нейронам.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

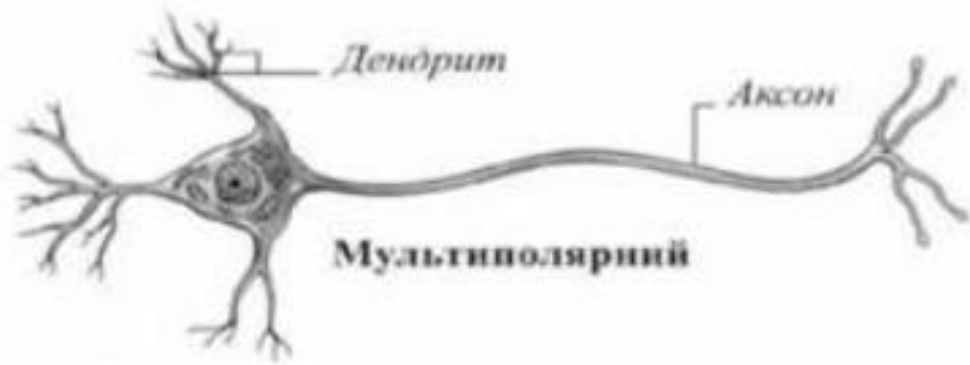


Рис. 1.3 - Мультиполярний нейрон [3]

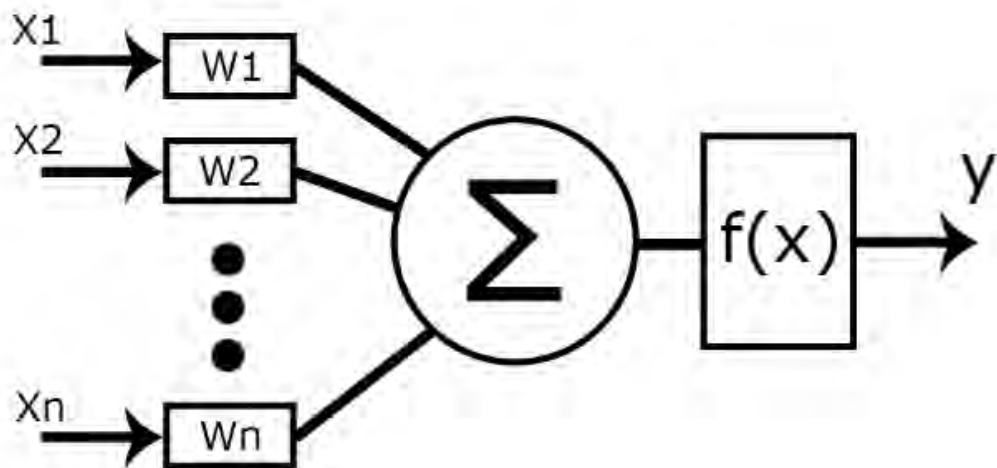


Рис. 1.4 - Нейрон Маккаллоха-Пітца [4]

Ваги нейронних мереж є параметрами, які визначають силу впливу вхідних сигналів на роботу нейрону. Вони встановлюються під час навчання мережі і визначають, як сильно кожен вхідний сигнал впливає на вихідний сигнал нейрону. Метафорична ілюстрація роботи нейронів, та вплив вагів нейронів на систему, що зображено на рис. 1.5[4]. Отже, нейрон пропустить

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

сигнал далі по «аксону», коли на його «дендрит» прийде сигнал більше, ніж «ваги» цього нейрону, сила сигналу визначається за різницею вхідного сигналу та порогового значення.

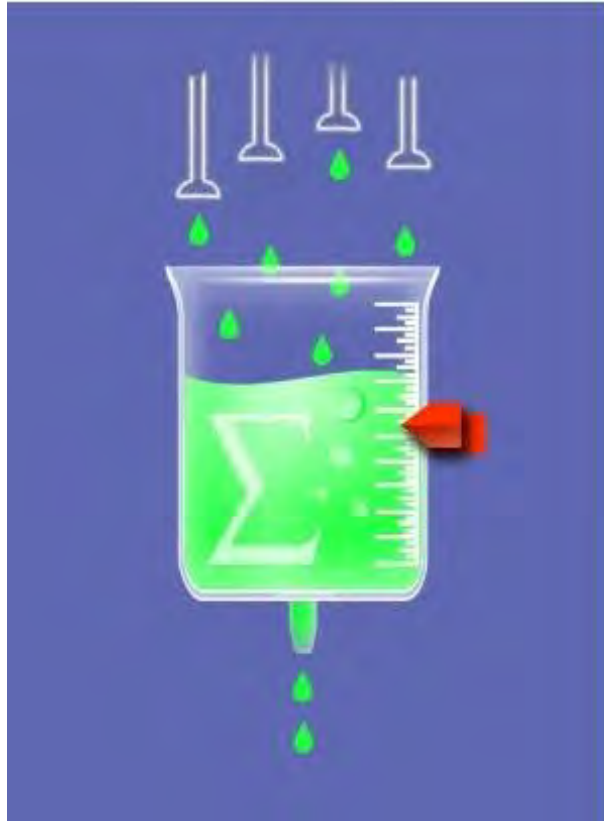


Рис. 1.5 - Ілюстрація роботи нейрона [4]

Нейронні мережі мають різні види, які використовуються для різних завдань машинного навчання та обробки даних. Основні види нейронних мереж включають:

1. Одношаровий перцептрон. Проста нейронна мережа, що складається з одного шару нейронів. Використовується для простих класифікаційних задач.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

2. Багатошаровий перцептрон. Мережа з одним або кількома прихованими шарами нейронів. Часто використовується для складних класифікаційних та регресійних задач.
3. Рекурентна нейронна мережа (RNN). Мережа, яка має циклічні зв'язки між нейронами, дозволяючи враховувати контекст та послідовність даних. Використовується для обробки послідовних даних, таких як мова та часові ряди.
4. Згорткова нейронна мережа (CNN). Мережа, спеціалізована на обробці зображень. Вона використовує спеціальні згорткові шари для виявлення локальних ознак у зображенні та пулінгові шари для зменшення розміру зображення. CNN широко використовується в задачах комп'ютерного зору, таких як виявлення об'єктів та класифікація зображень.

Так як у попередньо розглянутих прототипах для навчання нейронних мереж переважно використовують CNN мережі, розглядати інші види нейронних мереж для автоматизованої фотометричної системи буде дуже недоречно.

CNN (Convolutional Neural Network)

CNN (Convolutional Neural Network), або згорткова нейронна мережа, є одним з основних типів нейронних мереж, які використовуються для обробки і аналізу зображень та відео. Вона є ефективним інструментом для виявлення патернів та особливостей у вхідних даних, що дозволяє їй впевнено вирішувати завдання класифікації, локалізації та семантичного сегментування зображень.

Основні компоненти CNN зображенні на рис. 1.6, а саме:

1. Згортковий шар (Convolutional Layer). Цей шар використовує фільтри, що рухаються по вхідному зображенню для виконання згортки. Згортка

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

полягає в обчисленні поточного значення пікселя шляхом по елементного множення фільтра на відповідну область вхідного зображення і підсумовування результатів. Цей процес допомагає виділити різні особливості і структури на зображенні.

2. Шар пулінгу (Pooling Layer). Цей шар використовується для зменшення просторових розмірів представлення зображення, зберігаючи при цьому ключові особливості. Зазвичай використовується пулінг методу "max pooling", де відбирається максимальне значення з кожної області. Це дозволяє зменшити кількість параметрів і обчислювальну складність моделі.
3. Повно зв'язаний шар (Fully Connected Layer). Цей шар приймає векторизоване представлення особливостей з попередніх шарів і виконує класифікацію або регресію. Кожен нейрон у цьому шарі пов'язаний з кожним нейроном попереднього шару.
4. Функції активації (Activation Functions). Функції активації, такі як ReLU (Rectified Linear Unit), Sigmoid або Tanh, застосовуються після кожного шару для введення нелінійності в мережу. Вони допомагають моделі вирішувати складніші задачі та покращують здатність моделі до нелінійної апроксимації.

CNN має кілька повторюваних блоків згорткових шарів та шарів пулінгу, за якими слідує повно-зв'язаний шар для класифікації. Ця архітектура дозволяє моделі автоматично вивчати різні рівні абстракції від простих функцій до складних особливостей зображень.

Використання згорткових шарів та пулінгу допомагає зменшити кількість параметрів, покращити ефективність обчислень та робить модель більш інваріантною до зсуву та масштабування зображень. Також вона може використовуватися в поєднанні з іншими типами шарів, такими як рекурентні шари, для рішення складних завдань обробки послідовностей.

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

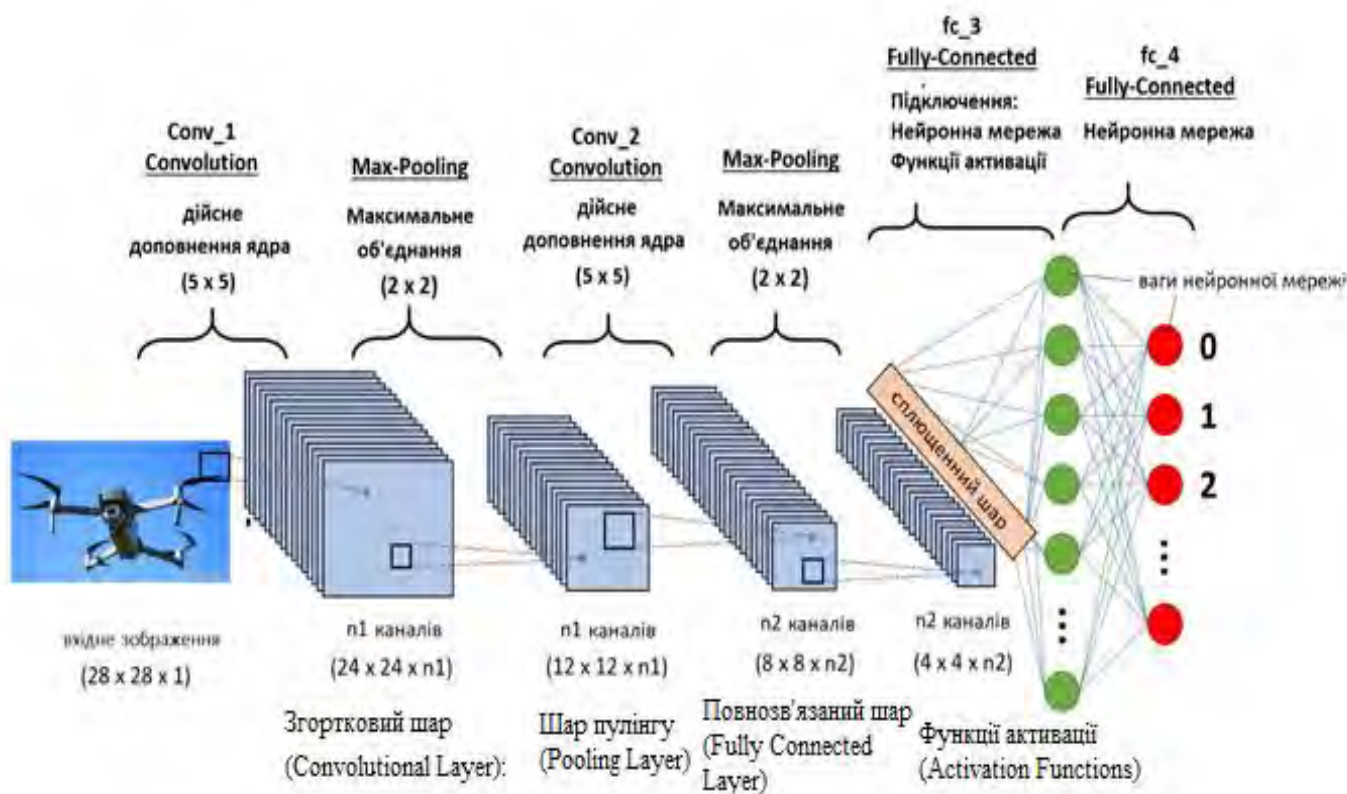


Рис. 1.6. - Згорткова нейронна мережа

Підходи до автоматизованих алгоритмів обробки зображень для комп'ютерного зору

Щодо підходів автоматизації комп'ютерного зору, двом з них варто звернути увагу: Hough Transform та Template Matching.

Hough Transform

Перетворення Хафа (Hough Transform) є одним з основних методів обробки зображень для розпізнавання різних форм та об'єктів. У його класичному застосуванні перетворення Хафа спрямоване на виявлення прямих ліній на зображеннях, що може бути використано для пошуку дорожніх

розміток в безпілотних автомобілях, виробничих маркерів для промислових робіт та для визначення перспективи простору на зображенні.

Алгоритм:

1. Визначення діапазону значень для ρ та θ :

- Діапазон для θ : $[0, 180]$ градусів;
- Діапазон для ρ : $[-d, d]$, де d - довжина діагоналі зображення ребра.

2. Опис фігури за допомогою рівняння:

- Для прямої: Рівняння:

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta), \quad (1)$$

де ρ - відстань від найближчої точки прямої до початку координат;

θ - кут між віссю X і прямою в точці перетину з початком координат.

3. Створення двовимірного масиву під назвою "акумулятор":

- Розміри: відповідно кількості параметрів у рівнянні;
- У випадку прямої: двовимірний масив.

4. Заповнення акумулятора:

- Початкове значення: нуль для всіх елементів масиву;
- Кожна пряма у просторі Хафа відповідає своїм параметрам у рівнянні.

5. Виявлення зображення границь (edge image):

- Використання алгоритму виявлення ребра для створення зображення границь.

Щоб отримати зображення границь (edge image), можна використати алгоритм виявлення ребер, такий як алгоритм Кенні або алгоритм Собеля.

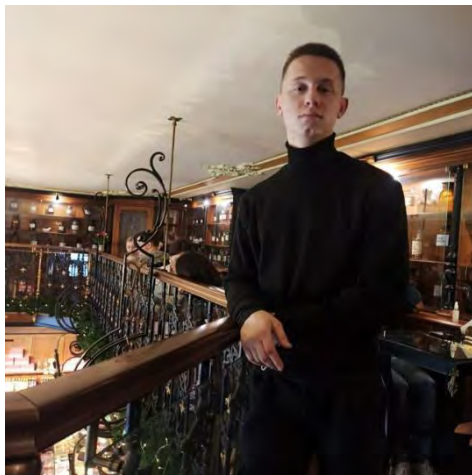
					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

Відповідно до алгоритму Кенні, наступні кроки можуть бути виконані для отримання зображення границь:

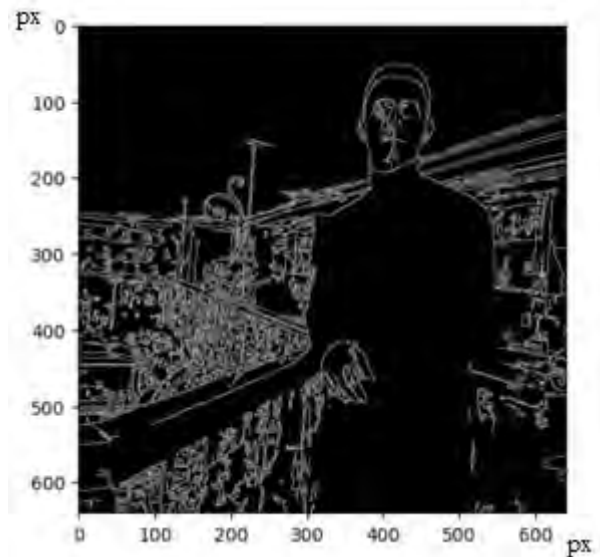
1. Перетворення вхідного зображення в градації сірого (використовуючи, наприклад, перетворення відтінків сірого чи середнє значення RGB-каналів).
2. Застосування фільтра Собеля для виявлення вертикальних та горизонтальних границь. Фільтр Собеля використовує дві матриці ядра для обчислення градієнта яскравості пікселів.
3. Обчислення апроксимації градієнта яскравості для кожного пікселя на зображенні. Це включає обчислення значень градієнта по горизонталі та вертикалі, а потім обчислення загального градієнта та напрямку.
4. Застосування порогового значення до значень градієнта, щоб відрізнити границі від шуму. Пікселі, які перевищують порогове значення, вважаються пікселями границь, а інші пікселі - фоном.
5. Застосування операції неперервності, такої як оператор нон-максимуму, для видалення непотрібних пікселів границь та тонких ліній.
6. Опціонально: застосування гістерезисного визначення порогів для з'єднання розрізаних границь та встановлення кінцевих границь.

Приклад реалізації зображення границь зображено на рис. 1.7.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						27
Зм.	Арк.	№ докум.	Підпис	Дата		



а)



б)

Рис. 1. 7: а) - оригінальне зображення, б) - зображення границь.

Для кожного пікселя на зображенні ребра перевіряється, чи є піксель пікселем ребра. Якщо це так, необхідно пройтись по всіх можливих значеннях θ , обчислити відповідне ρ , знайти індекси θ та ρ в акумуляторі і збільшити значення акумулятора для цих пар індексів.

Отримані координати пікселя ребра, наприклад, (x, y) , обробляються наступним чином для всіх можливих значеннях θ , кроком кроку (наприклад, від 0 до 180 градусів з певним кроком) обчислюється відповідне значення ρ за допомогою формули:

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta), \quad (2)$$

де (x, y) - отримані координати пікселя ребра;

θ - кут між віссю X і прямою в точці перетину з початком координат;

ρ - відстань від найближчої точки прямої до початку координат.

Розраховуються індекси θ та ρ в акумуляторі, використовуючи квантування діапазону значень ρ та θ . Індекс θ буде відповідати значенню θ , а індекс ρ буде відповідати квантованому значенню ρ .

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

Збільшують значення акумулятора для цих пар індексів, оновлюючи кількість голосів (або вагу) цієї лінії в акумуляторі.

Таким чином, кожен піксель ребра сприяє до підсумкового голосу кожної можливої лінії, яка може проходити через цей піксель. Кількість голосів для кожної лінії залежить від кількості пікселів ребра, через які вона проходить.

Перебираючи всі значення в акумуляторі, якщо значення перевищує певний поріг, отримані індекси ρ та θ , можна перетворити назад за формулою:

$$y = ax + b. \quad (3)$$

Під час процесу виявлення ліній на зображенні, жовті точки в просторі Хафа свідчать про наявність ліній, які представлені парою значень θ і ρ , що зображено на рис. 1.8. Ці точки вказують на потенційну присутність ліній у вихідному зображенні.

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

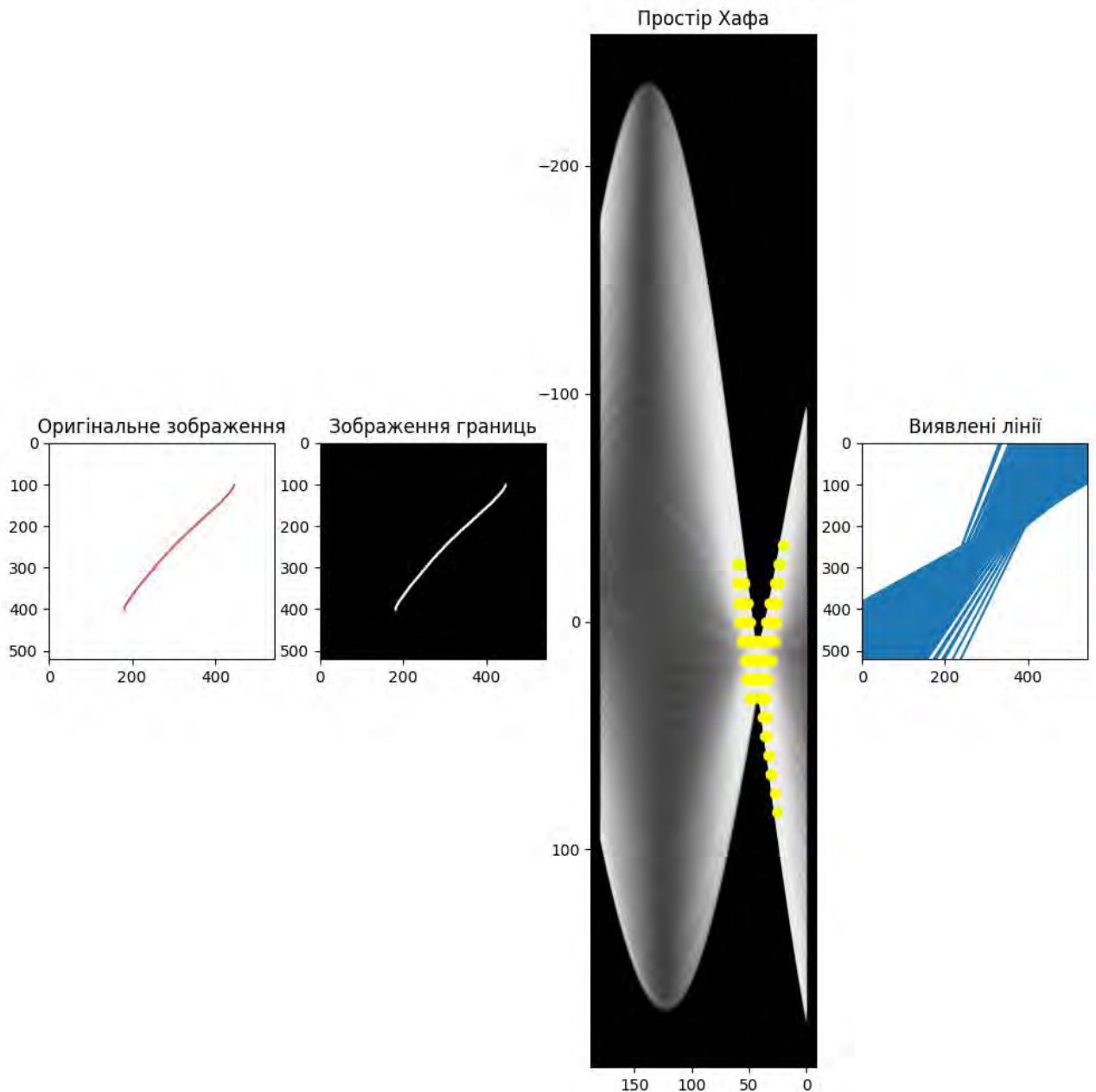


Рис. 1.8 - Зображення простору Хафа

Template Matching

Template Matching - це метод комп'ютерного зору, який шукає частини зображення, найбільш схожі на шаблон, який ви шукаєте. Його модифіковані варіанти дозволяють знайти шаблон навіть при різній яскравості, куті обертання та розмірі.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Метод шаблонного співставлення використовується для поліпшення точності виявлення об'єктів на зображенні. Цей метод ґрунтується на порівнянні шаблону з різними частинами зображення з метою знайти найбільш схожі частини. Ідея полягає у тому, що шаблон накладається на різні частини зображення і обчислюється ступінь подібності між накладеним шаблоном і підлеглою частиною зображення. Ця ступінь подібності виражається через коефіцієнт кореляції. Високі значення коефіцієнта кореляції вказують на те, що шаблон і частина зображення є схожими.

Метод ґрунтується на досить простій концепції накладання шаблону на зображення у різних позиціях. Під час накладання розраховується подібність між накладеним шаблоном і зображенням, на якому він накладений. Ступінь "подібності" або кореляції зображень виражається через перехресну кореляцію. Сама перехресна кореляція - це простий сумарний добуток відповідних пікселів зображень.

Нормалізована перехресна кореляція є кращим способом розрахунку "подібності" зображень, оскільки її результати не залежать від змін яскравості зображень. Конкретно, якщо зображення, на якому ви хочете знайти шаблон, має велику яскравість, нормалізована перехресна кореляція розраховує результати у межах [-1, 1], де повністю ідентичні елементи сходяться до 1, а в протилежному випадку - до -1.

Нормалізована перехресна кореляція обчислюється за наступною формулою:

$$NCC(im_1, im_2) = \frac{1}{N_{\sigma_1, \sigma_2}} \sum_{x,y} (im_1(x, y) - \bar{im}_1) \cdot (im_2(x, y) - \bar{im}_2), \quad (4)$$

де im_1 та im_2 - зображення та шаблон відповідно;

μ_1 та μ_2 - середні значення пікселів зображення та шаблону відповідно;

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

σ_1 та σ_2 - стандартні відхилення пікселів зображення та шаблону відповідно;

N - загальна кількість пікселів у шаблоні.

Отримані значення нормалізованої перехресної кореляції вказує на ступінь схожості між шаблоном і частиною зображення, де високі значення вказують на високу подібність.

Розбираючи приклад з використанням методу Template Matching, маємо зображення з випадковим розташуванням кругів різного розміру та кольору, що зображено на рис. 1.9.

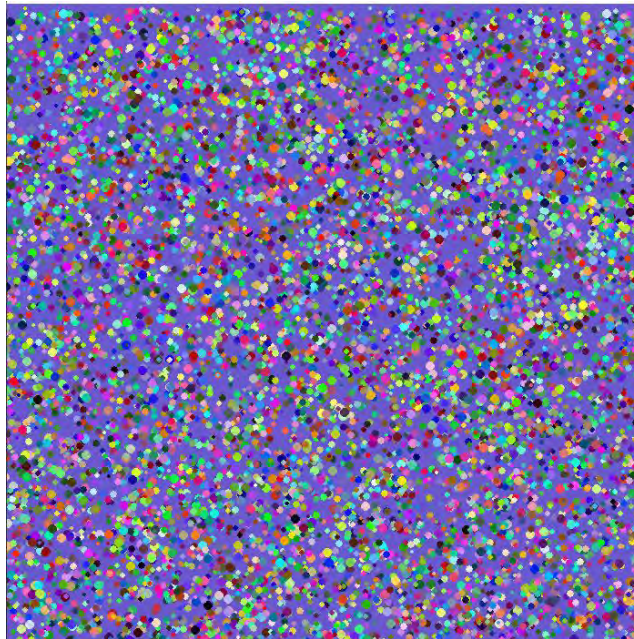
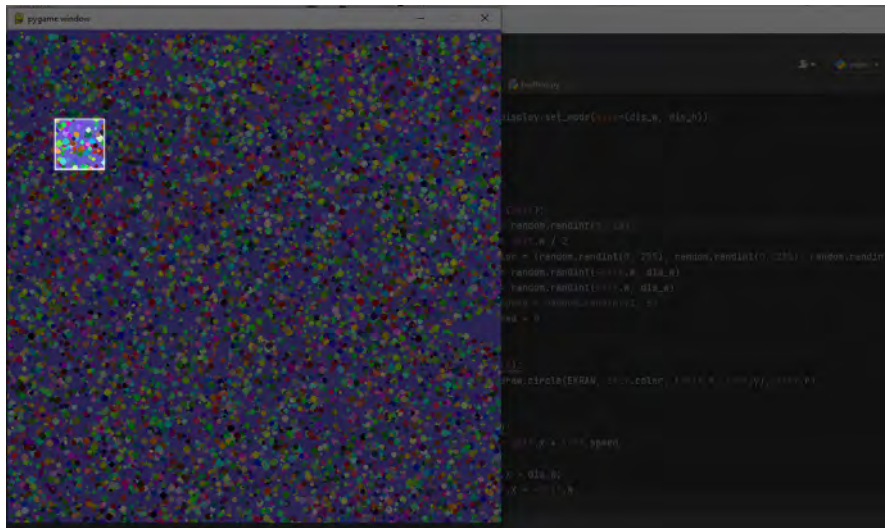


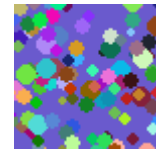
Рис. 1.9 - Оригінальне зображення

Для прикладу необхідно мати якийсь фрагмент з оригінального зображення, як на рис. 1.10.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						32
Зм.	Арк.	№ докум.	Підпис	Дата		



а)



б)

Рис. 1.10: а) - Розташування фрагменту який необхідно знайти б) - Фрагмент який необхідно знайти

Після цього застосовується метод Template Matching до кожного вирізаного фрагмента для виявлення шаблону. Проводиться аналіз, що включає порівняння фрагменту з фактичними позиціями та розмірами кругів на вихідному зображенні. Результат аналізу методом зображений на рис. 1.11.

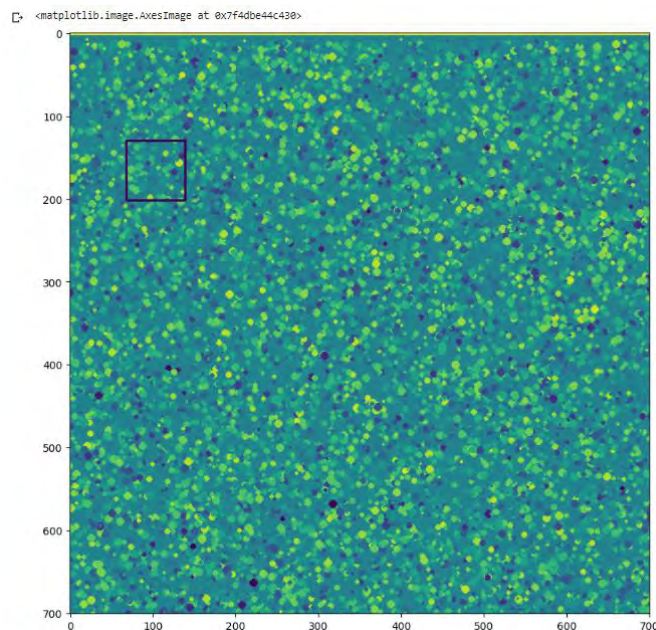


Рис. 1.11 - Результат Template Matching

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

Плюси та недоліки

Плюси:

- Метод Template Matching є простим у реалізації та розумінні;
- Він дозволяє знаходити шаблони навіть при різній яскравості, різних розмірах та куті повороту;
- Результати методу можна графічно перевірити, що сприяє валідації.

Недоліки:

- Метод може бути схильний до помилкових виявлень, особливо якщо об'єкти мають схожу яскравість, але різну форму;
- Якщо на зображенні присутній шум або змінюється яскравість, можуть виникати неточності у виявленні шаблону;
- Метод Template Matching може бути витратним за часом обчислення, особливо при великому розмірі зображення та шаблону.

В цілому, метод Template Matching є корисним інструментом для виявлення шаблонів на зображеннях, але його ефективність може залежати від конкретних умов застосування та власних цілей.

1.3 Огляд фреймворків

YOLO (You Only Look Once)

YOLO (You Only Look Once) є популярною архітектурою глибокого навчання для об'єктної локалізації та класифікації об'єктів у зображеннях. Основна ідея YOLO полягає в тому, щоб одночасно прогнозувати багато

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

областей (bounding boxes) та відповідних класів об'єктів у цих областях, використовуючи одну пряму прохідну нейронну мережу.

Основні характеристики YOLO:

1. Швидкодія: YOLO є швидким методом локалізації об'єктів, оскільки виконує прогнози для всього зображення в одній подачі. Це відрізняє його від інших архітектур, які використовують підходи з областями зацікавлення (Region Proposal), такі як R-CNN, які вимагають багато кроків та обчислень для кожного об'єкта.
2. Багатомасштабна архітектура: YOLO використовує кілька шарів з різною роздільною здатністю для виявлення об'єктів на різних масштабах. Це дозволяє моделі ефективно прогнозувати як великі, так і малі об'єкти на зображенні.
3. Використання конволюційних шарів: Архітектура YOLO базується на використанні конволюційних нейронних мереж для виявлення ознак у зображенні та локалізації об'єктів.
4. Помноження характеристик (Feature Pyramid Network): YOLO використовує Feature Pyramid Network (FPN), щоб впоратися з проблемою виявлення об'єктів різного масштабу. FPN дозволяє моделі виявляти як великі, так і малі об'єкти шляхом аналізування різних рівнів деталей зображення.
5. Використання Grid-Cell: YOLO розбиває вхідне зображення на сітку з фіксованим розміром та прогнозує області (bounding boxes) та класи об'єктів для кожної комірки сітки. Це дозволяє моделі ефективно виявляти об'єкти у всьому зображенні та точно локалізувати їх.
6. Non-Maximum Suppression: Після прогнозування багатьох областей, YOLO використовує алгоритм Non-Maximum Suppression для фільтрації накладених областей та вибору найбільш ймовірної області для кожного об'єкта.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

Загалом, YOLO є ефективною та швидкою архітектурою для виявлення та класифікації об'єктів у зображеннях. Вона знаходить застосування в багатьох сферах, включаючи комп'ютерне зорове сприйняття, фотометричні системи автономну навігацію, відеоспостереження та багато інших.

OpenCV (Open Source Computer Vision Library)

OpenCV (Open Source Computer Vision Library) є потужною бібліотекою комп'ютерного зору, яка надає широкий спектр функцій для обробки зображень, включаючи розпізнавання об'єктів (object detection). Існує кілька способів використання OpenCV для реалізації системи розпізнавання об'єктів. Одним з популярних підходів є використання методів, які базуються на машинному навчанні, зокрема нейронних мереж.

Зазвичай використовуються попередньо навчені моделі, які були навчені на великому наборі даних, наприклад, моделі з серії YOLO (You Only Look Once) або SSD (Single Shot MultiBox Detector). Ці моделі містять ваги, що були оптимізовані для розпізнавання об'єктів на зображеннях.

TensorFlow та Keras

TensorFlow та Keras є двома популярними бібліотеками для розробки та навчання нейронних мереж.

TensorFlow - це відкрите програмне забезпечення для чисельних обчислень та машинного навчання, розроблене компанією Google. Вона надає потужний фреймворк для побудови та навчання різних моделей машинного навчання, у тому числі нейронних мереж. TensorFlow забезпечує гнучкість та ефективність завдяки використанню графових обчислень та автоматичного диференціювання.

Основні можливості TensorFlow включають:

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Визначення та побудова моделей. TensorFlow дозволяє визначати складні моделі нейронних мереж за допомогою високорівневого API `tf.keras` або низькорівневих операцій тензорів.
2. Автоматичне диференціювання. TensorFlow автоматично обчислює градієнти для здійснення оптимізації моделі під час навчання.
3. Розпаралелене обчислення. TensorFlow може використовувати різні пристрої, такі як CPU або GPU, для розпаралеленого обчислення, що дозволяє прискорити процес навчання та застосування моделі.

Keras - це високорівневий API для розробки нейронних мереж, який працює на основі TensorFlow (а також може використовуватися з іншими бекендами, такими як Theano або CNTK). Keras спрощує процес визначення, навчання та застосування моделей, роблячи його більш зрозумілим та доступним.

Основні особливості Keras:

1. Легкість використання. Keras надає простий та інтуїтивно зрозумілий інтерфейс для визначення та навчання моделей. Він дозволяє швидко створювати прототипи моделей без необхідності глибоких знань про внутрішню роботу нейронних мереж.
2. Модульність. Keras дозволяє будувати моделі за допомогою зручних блоків, таких як шари, функції активації та оптимізатори. Ви можете легко комбінувати ці блоки для створення складних моделей.
3. Підтримка мультибекендів. Keras підтримує використання різних бекендів, таких як TensorFlow, Theano або CNTK, що дозволяє гнучко працювати зі зручним API Keras та використовувати потужність різних фреймворків.

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

За замовчуванням Keras використовує TensorFlow як свій бекенд, тому ви можете використовувати всі можливості TensorFlow разом з простотою та зручністю Keras.

PyTorch

PyTorch є одним з провідних фреймворків для розробки нейронних мереж та проведення обчислень у галузі машинного навчання.

Особливості PyTorch:

1. Динамічний графік обчислень: PyTorch використовує динамічний графік обчислень, що дозволяє легко визначати та змінювати структуру моделі під час виконання. Це надає більшу гнучкість та зручність при розробці та відладці моделей.
2. Простота використання: PyTorch має простий та зрозумілий API, що спрощує процес визначення та навчання моделей. Його синтаксис наближений до стандартного Python, що робить роботу з фреймворком інтуїтивно зрозумілою для розробників.
3. Модульність: PyTorch побудований на основі модульної структури, де кожен компонент, такий як шари, функції активації та оптимізатори, може бути використаний окремо. Це дозволяє гнучко складати та змінювати моделі за потреби.
4. Підтримка GPU обчислень: PyTorch надає можливість використання графічних процесорів (GPU) для прискорення обчислень. Це особливо важливо при роботі з великими об'ємами даних та складними моделями.
5. Багатофункціональність: PyTorch має велику кількість вбудованих функцій та бібліотек для обробки даних, візуалізації, валідації моделей та інших завдань, пов'язаних з машинним навчанням.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

PyTorch є популярним вибором для дослідників та розробників завдяки своїй простоті, гнучкості та ефективності у роботі з нейронними мережами. Він надає широкі можливості для розробки та навчання моделей, а також для реалізації різноманітних алгоритмів у галузі машинного навчання.

1.4 Огляд застосунків для збору даних та розмітки

Для отримання якісної моделі комп'ютерного зору з великою кількістю даних варто використовувати наступні кроки та інструменти для збору даних. Для збору даних можна використовувати різні джерела, такі як відкриті набори даних, датасети, фото або відеозйомку. Також можна використовувати спеціалізовані застосунки для збору даних, які дозволяють автоматизувати процес.

Огляд деяких популярних застосунків для збору даних та розмітки в області комп'ютерного зору представлено нижче.

Kaggle

Kaggle є платформою для спільної роботи над проектами з машинним навчанням. Вона також надає доступ до великої кількості даних та розмітки для розв'язання завдань у сфері комп'ютерного зору. На Kaggle можна знайти різноманітні набори даних, як приклад зображено на рис. 1.12, датасети та розмітку, які можна використовувати для тренування моделей.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

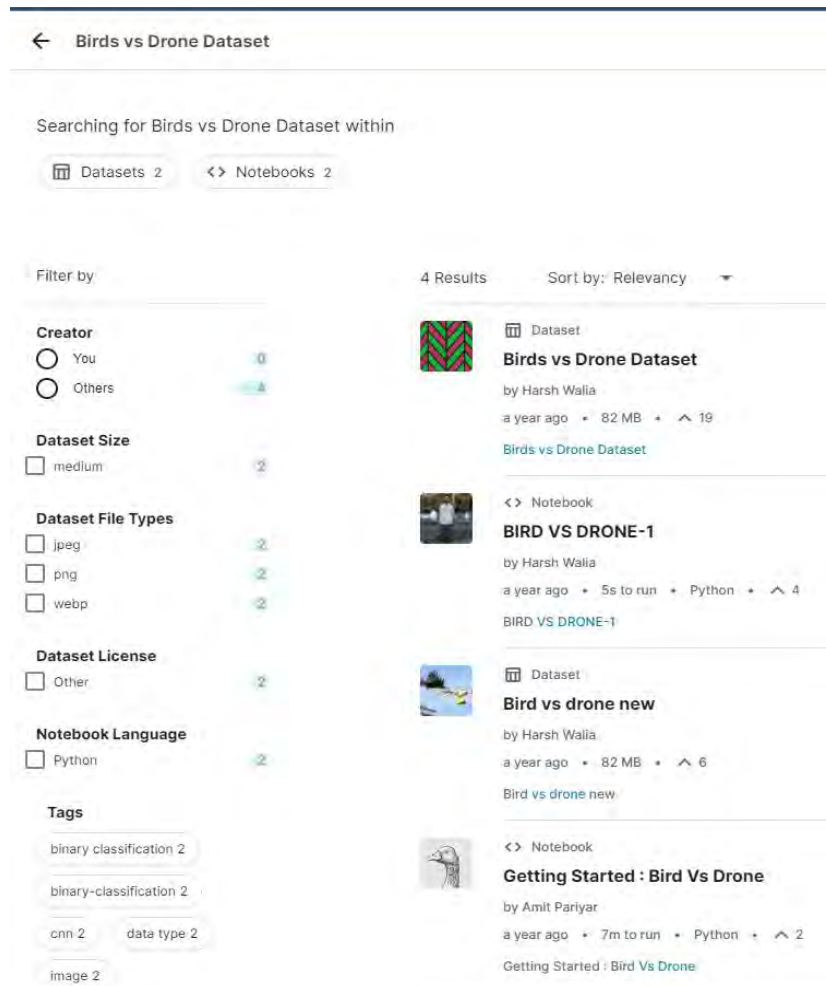


Рис. 1.12 - Kaggle

CVAT (Computer Vision Annotation Tool)

CVAT - це потужний веб-інтерфейс для розмітки зображень та відео. Він надає зручний інтерфейс для розмітки об'єктів, таких як області цікавості, прямокутники, полігони, ключові точки тощо, як приклад зображено на рис. 1.13. CVAT також підтримує розмітку видів об'єктів, доріжок та атрибутів. Він дозволяє командам ефективно співпрацювати над розміткою та збирати якісні дані для тренування моделей комп'ютерного зору.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ 40	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		

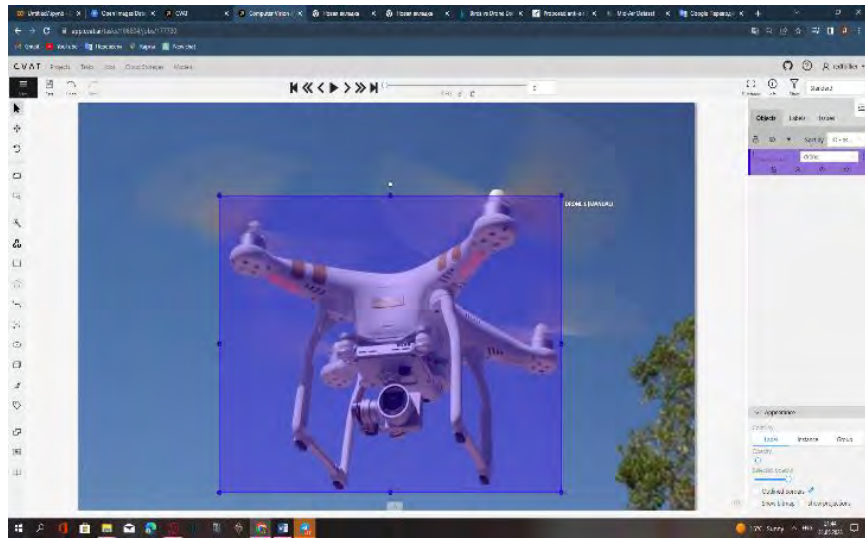


Рис. 1.13 - Computer Vision Annotation Tool. Розмітка даних

Roboflow

Roboflow - це інструмент для підготовки даних та розмітки у сфері комп'ютерного зору. Він надає різноманітні функції для завантаження, обробки, розмітки та підготовки даних для навчання моделей комп'ютерного зору. Roboflow підтримує різні формати даних, а також надає можливість автоматичної розмітки та аугментації даних. Він також пропонує інтеграцію з популярними фреймворками для навчання моделей, що зображено на рис. 1.14.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

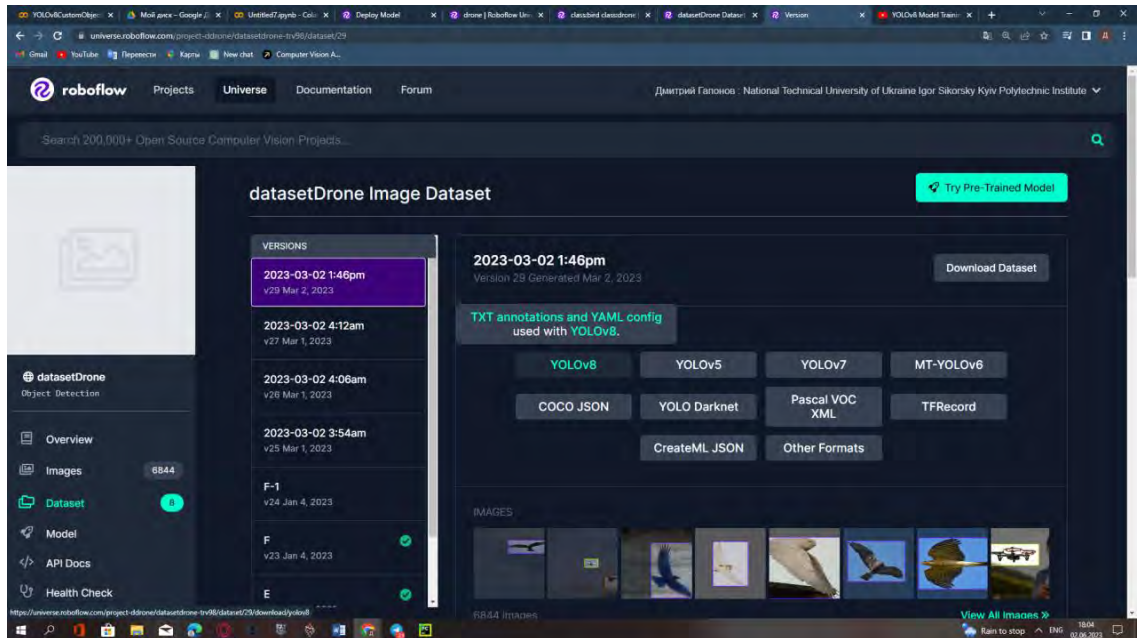


Рис. 1.14 - Датасет Roboflow

Ці застосунки є популярними серед дослідників, команд розробників та спеціалістів з обробки зображень для збору даних та розмітки, які є важливим етапом у розробці моделей комп'ютерного зору. Кожен з них має свої унікальні особливості та можливості, які можуть варіюватися в залежності від конкретних потреб та завдань роботи.

						Арк
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

ДПБ.ПМ-91.05.1760.01.000.ПЗ

Висновки до розділу 1

На підставі порівняння, можна зробити наступні висновки:

1. Системи класифікації. Найкращим варіантом є використання автоматизованої фотометричної системи з використанням комп'ютерної або серверної частини. Це дозволяє забезпечити точність та ефективність обробки даних.
2. Нейронні мережі. Згорткові нейронні мережі (Convolutional Neural Networks, CNN) є найкращим вибором для комп'ютерного зору, оскільки вони спеціалізуються на обробці зображень і мають високу ефективність у розпізнаванні об'єктів.
3. Фреймворки. За фреймворки для комп'ютерного зору найкращими варіантами є YOLOv8 (You Only Look Once) і OpenCV (Open Source Computer Vision Library). YOLO є швидким та точним фреймворком для об'єктного розпізнавання, а OpenCV надає багато функцій для обробки зображень та роботи з даними комп'ютерного зору.
4. Застосунки для збору даних та розмітки. Roboflow є кращим вибором для збору даних та розмітки. Він надає зручний інтерфейс та автоматизовані інструменти для збору, розмітки та підготовки даних перед тренуванням моделі комп'ютерного зору.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

2. ПРАКТИЧНА ЧАСТИНА

2.1 Структурна схема та алгоритм роботи системи.

Структурна схема автоматизованої фотометричної системи зображена на рис. 2.1 і складається з таких елементів як: IP камери (максимальна кількість до 8 штук), комутатор, ПК, монітор, джерело живлення.

IP камери: У системі присутні декілька IP камер, які призначені для збору зображень або відео з об'єктів, що аналізуються. Ці камери можуть бути розташовані в різних місцях або точках спостереження. Вони підключені до джерела живлення для постійного функціонування.

Комутатор: IP камери підключені до комутатора, який є мережевим пристроєм для забезпечення комунікації між камерами та ПК. Комутатор має порти POE для підключення IP камер і один або декілька портів для підключення до ПК або мережі. **Джерело живлення POE:** Комутатор забезпечує електроенергію для IP камер, щоб забезпечити їхню роботу.

ПК є центральною частиною системи і виконує обробку та аналіз зібраних зображень або відео. Він підключений до комутатора, щоб отримувати дані від IP камер. ПК може мати достатню обчислювальну потужність та пам'ять для виконання фотометричних алгоритмів та обробки даних.

Монітор використовується для відображення результатів фотометричного аналізу. Він підключений до ПК і дозволяє користувачу переглядати та оцінювати отримані дані або зображення. Монітор може мати високу роздільну здатність та кольорову точність для точного відображення результатів аналізу.

Огляд алгоритму роботи системи представлено на рис. 2.2.

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

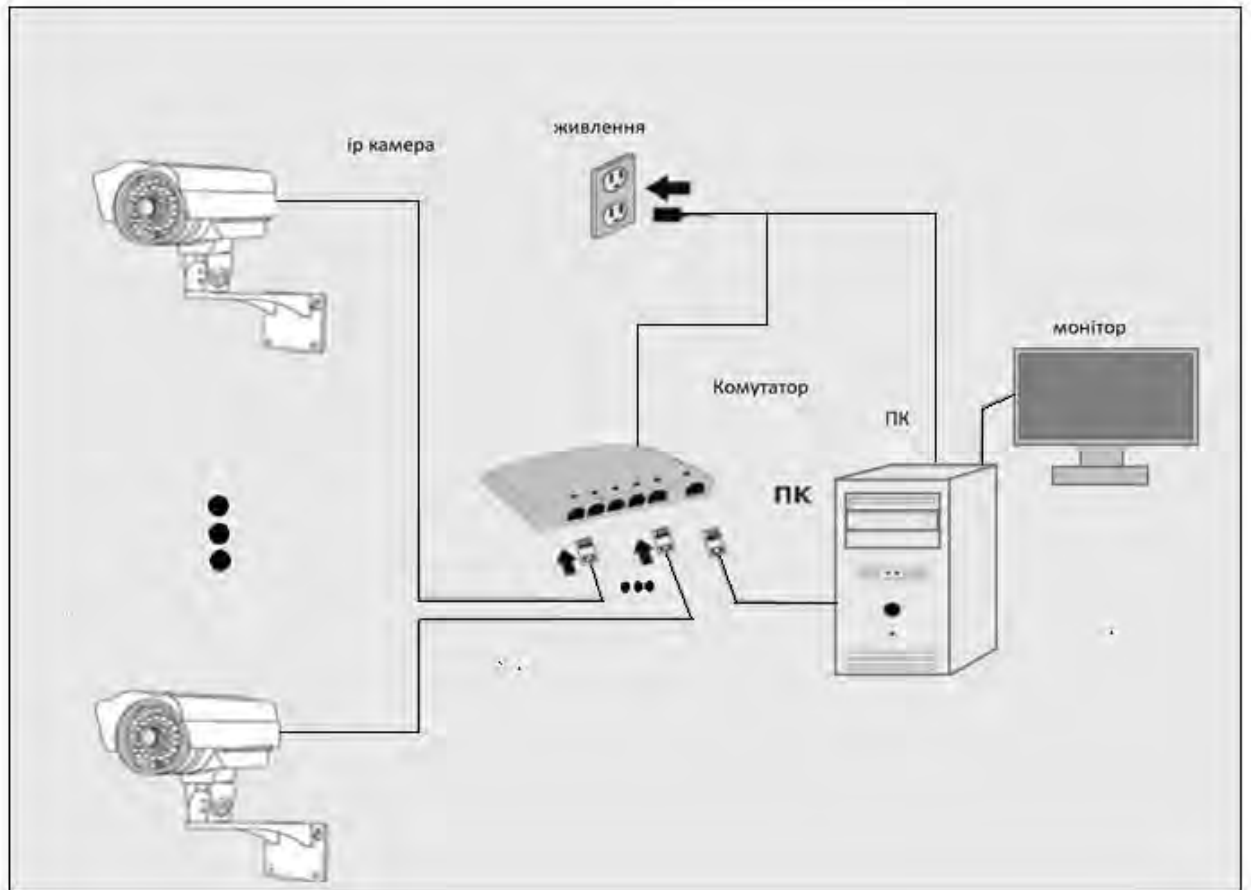


Рис. 2.1 - Структурна схема

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

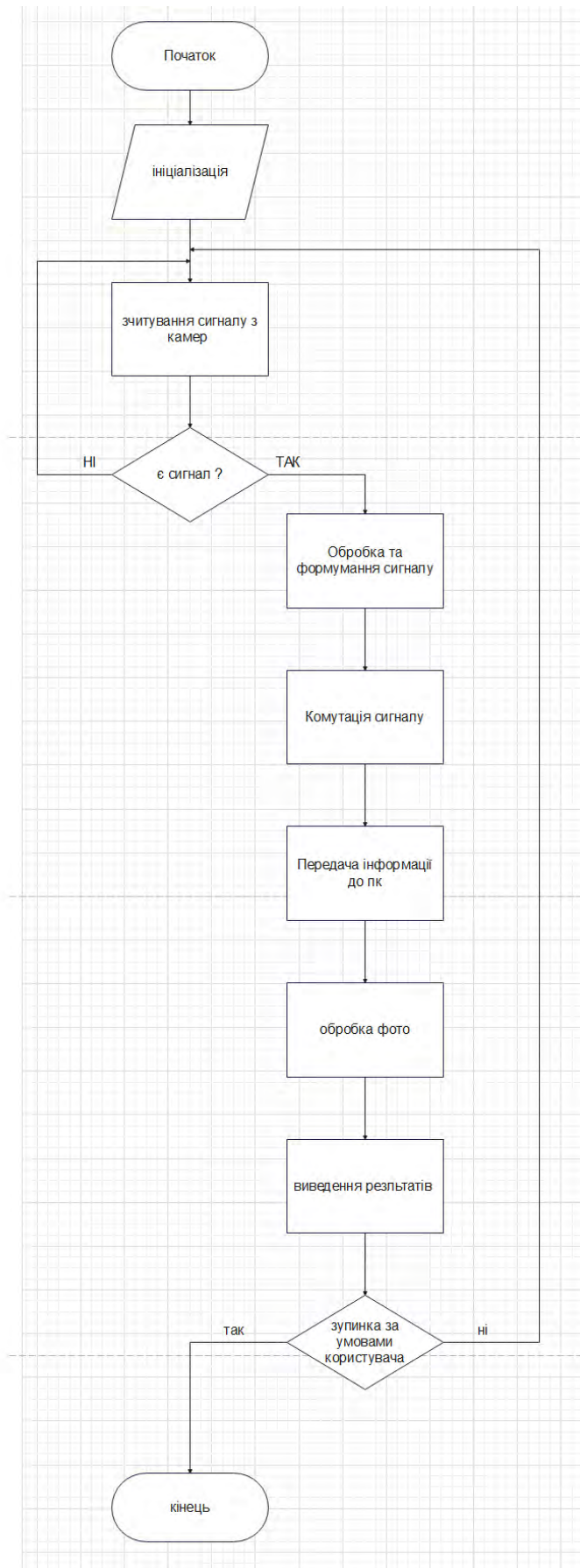


Рис. 2.2 - Алгоритм роботи автоматизованої фотометричної системи для розпізнавання об'єктів

2.2 Вибір технічних компонентів

Вибір камери

У якості камери обираємо модель VL-8739D [5], яка є надчутливою камерою нагляду з роздільною здатністю 600 ТВЛ. Ця камера володіє декількома технологіями, які забезпечують високу якість зображення та покращують її функціональність.

VL-8739D - високочутлива камера спостереження з роздільною здатністю 600 ТВЛ і функціями SENSE-UP та 3D-NR, що зображена на рис. 2.3.



Рис. 2.3 - Камера VL-8739D [5]

Технічні характеристики[5]:

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

- Сенсор зображення SONY 1/3" Super HAD II* ICX-639BKA;
- Роздільна здатність камери 600;
- Ефективна кількість пікселів (Ш x В) ~440К (752 x 582);
- Мінімальна освітленість (режим "день") 0.2 лк / F2.0;
- Мінімальна освітленість (режим "ночі") 0.05 лк / F2.0;
- Мінімальна освітленість (режим "SENS-UP") 0.005 лк / F2.0;
- Співвідношення сигнал/шум, не менше, дБ 50;
- Процесор обробки відео ISP-VII;
- Електронний затвор, с 1/50 - 1/100000;
- Відеовихід 1.0 В, 75 Ом, композитний;
- Напруга живлення через РОЕ, В 10.8 - 13.2 (постійний струм);
- Споживання струму, не більше, мА 110;
- Максимальна потужність (Pmax) відповідає максимальному значенню напруги та струму 1.452 мВ
- Мінімальна потужність (Pmin) відповідає мінімальному значенню напруги та струму 1.188 мВ
- Тип об'єктива: фіксований фокус;
- Фокусна відстань об'єктива (мм); світлосила f=3.6 мм; F=2.0;
- Додатковий об'єктив: фокусна відстань (мм); світлосила f=4 / 6 / 8 / 12 / 16 мм; F=2.0;
- Режим D-WDR (розширений динамічний діапазон);
- Режим 2D-NR (2D шумоподавлення);
- Режим 3D-NR (3D шумоподавлення за Гаусом);

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

- Режим HLC (приглушення надлишкової яскравості);
- Режим SenseUP (підвищення чутливості) + (від 2 до 256 разів);
- Діапазон робочих температур, град. С -25 ... +50 (від джерела безперебійного живлення);
- Ступінь захисту корпусу згідно зі стандартом ГОСТ 14254-96 (МІК 529-89) IP-65;
- Габаритні розміри, мм 78 x 150 (діаметр x довжина);
- Маса нетто (у повній комплектації), не більше 550 г.

Також габаритні креслення ІР камери зображено на рис. 2.4.

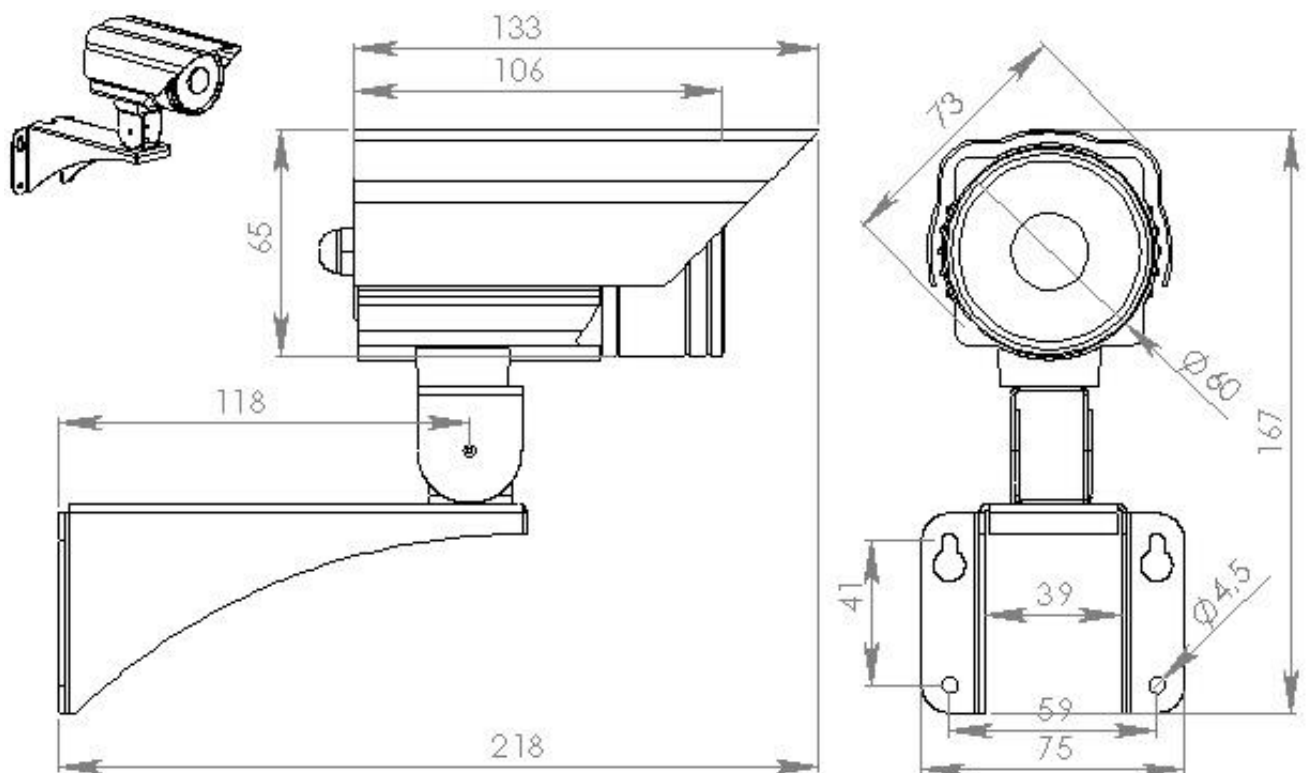


Рис. 2.4 - Габаритні креслення VL-8739D[5]

						Арк
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

ДПБ.ПМ-91.05.1760.01.000.ПЗ

Вибір комутатора

У якості POE комутатора обираємо світч 48v 10 портів (8poe+2uplink), що дає можливість підключення до 8 камер одночасно, що зображено на рис. 2.5.



Рис. 2.5 - POE комутатор 48V 100Mbps [6]

POE комутатор 48V 100Mbps характеристики [6]:

- Блок живлення вбудований в корпус: 52V / 2,3A / 120Вт, шнур живлення у комплекті;
- 2 порти Uplink (без POE) для підключення наприклад роутера + відеореєстратора 8 портів POE для підключення пристроїв POE з живленням;
- PoE-контролер може автоматично визначити та забезпечити живлення всіх пристроїв IEEE 802.3af/at (PD);
- Підтримує живлення по кабелю Cat.5 Ethernet бездротових точок доступу, мережевих камер відеоспостереження та IP-телефонів;

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

- Комутатор некерований, ніяких налаштувань немає (принцип Plug&Play – увімкнув і все працює);
- Потужність кожного порту PoE(af) до 15.4Вт або PoE+(at) до 30Вт;
- Звичайна IP камера споживає в середньому 5вт вдень / 8вт вночі;
- Хороший запас потужності, вистачає як для поворотних камер, так і для камер з посиленням нічним підсвічуванням;
- Габарити: 205x140x45мм.

2.3 Програмні компоненти

За допомогою Roboflow знайдено і завантажено готовий датасет з вже позначеними об'єктами дронів на рис. 2.6.

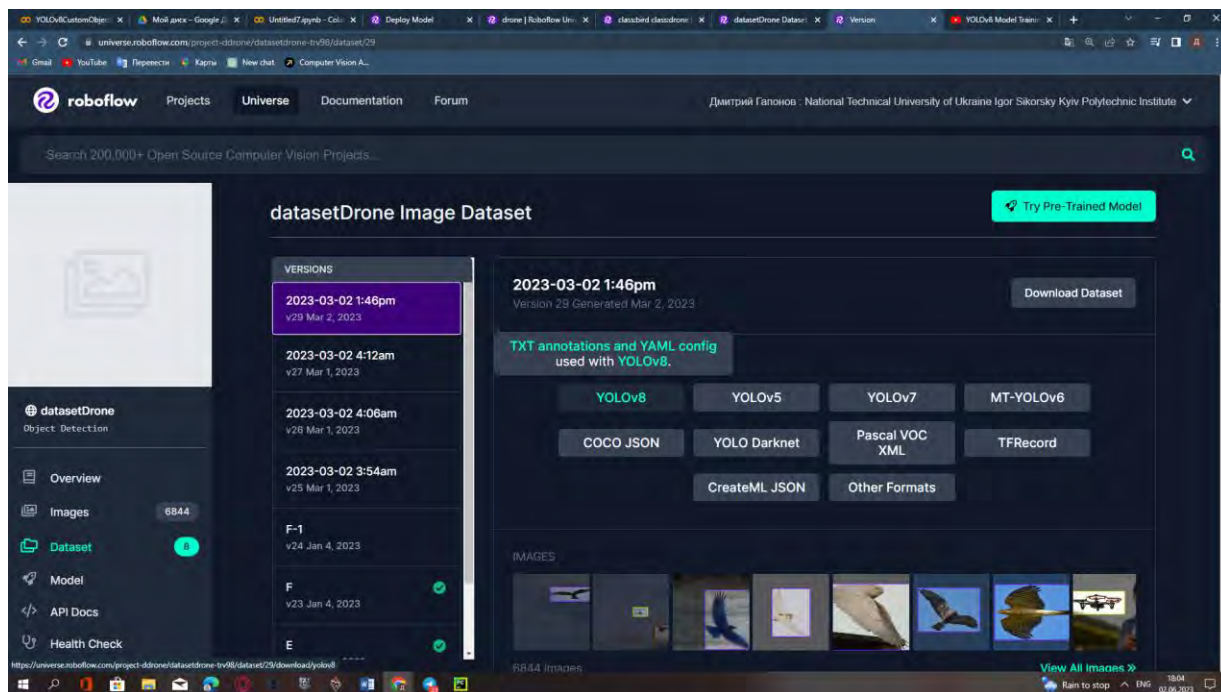


Рис. 2.6 - Завантаження датасету

						Арк
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

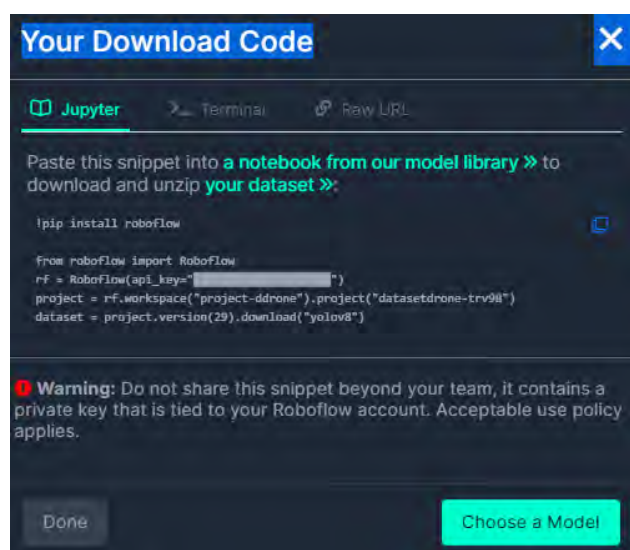
ДПБ.ПМ-91.05.1760.01.000.ПЗ

Для розв'язання проблеми розрізнення дронів і птахів у небі використано готовий датасет, який містить дані про різні об'єкти, що можуть зустрічатися у повітрі, крім дронів. Це включає такі об'єкти як птахи, літаки, повітряні кулі та інші. Цей датасет більше у 48 разів, ніж той, що ми змогли зробити самостійно.

Завдяки розширеному датасету навчено нашу нейронну мережу відрізняти дрони від інших об'єктів у повітрі. Це дозволить отримати більш точну та надійну класифікацію дронів під час тренування.

Однак, замість завантаження усього датасету на локальний комп'ютер, було використано функціональність платформи Roboflow у поєднанні з бібліотекою Ultralytics. Замість завантаження даних, конвертовано датасет у віртуальний ключ, який був доступний для використання безпосередньо в середовищі Ultralytics.

Цей підхід має переваги, так як було уникнено завантаження великого обсягу даних на наш локальний комп'ютер, що дозволило нам уникнути проблем з обмеженою пам'яттю та швидкістю запису. Також це зробило процес тренування більш зручним і ефективним. Приклад отриманого коду для завантаження даних зображено на рис. 2.7.



```
pip install roboflow

from roboflow import RoboFlow
rF = RoboFlow(api_key="")
project = rF.workspace("project-ddrone").project("datasetdrone-trv98")
dataset = project.version(29).download("yolov8")
```

Warning: Do not share this snippet beyond your team, it contains a private key that is tied to your Roboflow account. Acceptable use policy applies.

Рис. 2.7 - Код для завантаження датасету у середу розробки.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

Далі переходимо до тренування нейронної мережі, як зображено на рис. 2.8.

```
▼ завантажуюємо YOLO

✓ [3] from ultralytics import YOLO
import os
from IPython.display import display, Image
from IPython import display
display.clear_output()
!yolo mode=checks

▼ Тренуємо нейронну мережу для нашого завдання

# Встановлюємо бібліотеку Roboflow
!pip install roboflow

# Імпортуємо клас Roboflow з бібліотеки
from roboflow import Roboflow

# Ініціалізуємо об'єкт Roboflow і передаємо API-ключ
rf = Roboflow(api_key="CFD58sLmdkQJbjs05uwm")

# Вибираємо робочий простір та проект roboflow для завантаження даних у середовище
workspace_name = "project-ddrone"
project_name = "datasetdrone-trv98"
project = rf.workspace(workspace_name).project(project_name)

# Завантажуємо вказану версію датасету
dataset_version = 29
dataset_name = "yolov8"
dataset = project.version(dataset_version).download(dataset_name)
```

Рис 2.8 - Завантажуємо та тренуємо YOLOv8.

!pip install roboflow: Цей рядок встановлює бібліотеку Roboflow, яка необхідна для роботи з API Roboflow. Команда !pip install встановлює пакет з репозиторію PyPI.

From roboflow import Roboflow: Цей рядок імпортує клас Roboflow з бібліотеки Roboflow. Цей клас надає функціональність для роботи з API Roboflow.

rf = Roboflow(api_key="CFD58sLmdkQJbjsO5uwm"): Цей рядок створює об'єкт rf класу Roboflow і передає йому API-ключ, отриманий від Roboflow. Цей ключ дозволяє доступатися до проектів та датасетів на платформі Roboflow.

project = rf.workspace("project-ddrone").project("datasetdrone-trv98"): Цей рядок створює об'єкт project, який представляє проект на платформі Roboflow. Він використовується для роботи з датасетами, моделями та іншими аспектами проекту. В даному випадку, шлях до проекту вказаний як "project-ddrone" і назва датасету як "datasetdrone-trv98".

dataset = project.version(29).download("yolov8"): Цей рядок створює об'єкт dataset шляхом завантаження версії датасету з проекту. Версія датасету вказується як число 29, а назва моделі, яку ви хочете завантажити, вказується як "yolov8". Завантажений датасет можна використовувати для навчання моделі машинного навчання.

Після виконання коду програми, завантажено датасет з файлами, як на рис. 2.9.

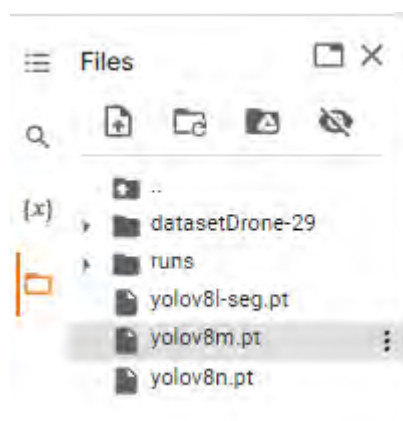


Рис. 2.9 - Завантажений датасет.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

Далі необхідно запуснути процес тренування нейронної мережі, як приклад на рис. 2.10.

▼ Запускаємо процес тренування

```
[ ] !yolo task=detect mode=train model=yolov8m.pt data={dataset.location}/data.yaml patience=10 imgsz=640
```

Рис. 2.10 – Тренування

Після того, як код програми виконався і було навчено нашу нейронну мережу, необхідно зберегти результати до гугл диску, що зображено на рис. 2.11:

```
[ ] # підключаємо гугл диск до системи
from google.colab import drive
drive.mount("/content/gdrive")

[1] # завантажуюмо результати на гугл диск
!scp -r /content/runs "/content/gdrive/My Drive/диплом"
```

Рис. 2.11 - Завантаження даних

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						55
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4 Математичне моделювання навчання нейронної мережі, огляд отриманих результатів.

Представленні графіки функції втрат, які відображаються під час тренування моделі за кількістю епох, зображені на рис. 2.12, надають важливу інформацію про прогрес і якість навчання.

Пояснення назв графіків:

- **train/box_loss**: Графік показує зміну значення функції втрат (loss) для локалізації об'єктів (bounding box) на тренувальному наборі даних. Чим нижче значення, тим краще модель виконує завдання локалізації;
- **train/cls_loss**: Графік відображає зміну значення функції втрат (loss) для класифікації об'єктів на тренувальному наборі даних. Цей графік вказує на те, наскільки добре модель визначає класи об'єктів;
- **train/dfl_loss**: Графік показує зміну значення функції втрат (loss) для додаткових дескриптивних атрибутів об'єктів (наприклад, орієнтація або поза) на тренувальному наборі даних. Це використовується у деяких моделях для вдосконалення точності опису об'єктів;
- **val/box_loss**: Аналогічно до **train/box_loss**, цей графік показує зміну значення функції втрат (loss) для локалізації об'єктів на валідаційному наборі даних;
- **val/cls_loss**: Аналогічно до **train/cls_loss**, цей графік відображає зміну значення функції втрат (loss) для класифікації об'єктів на валідаційному наборі даних;
- **val/dfl_loss**: Аналогічно до **train/dfl_loss**, цей графік показує зміну значення функції втрат (loss) для додаткових дескриптивних атрибутів об'єктів на валідаційному наборі даних;

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

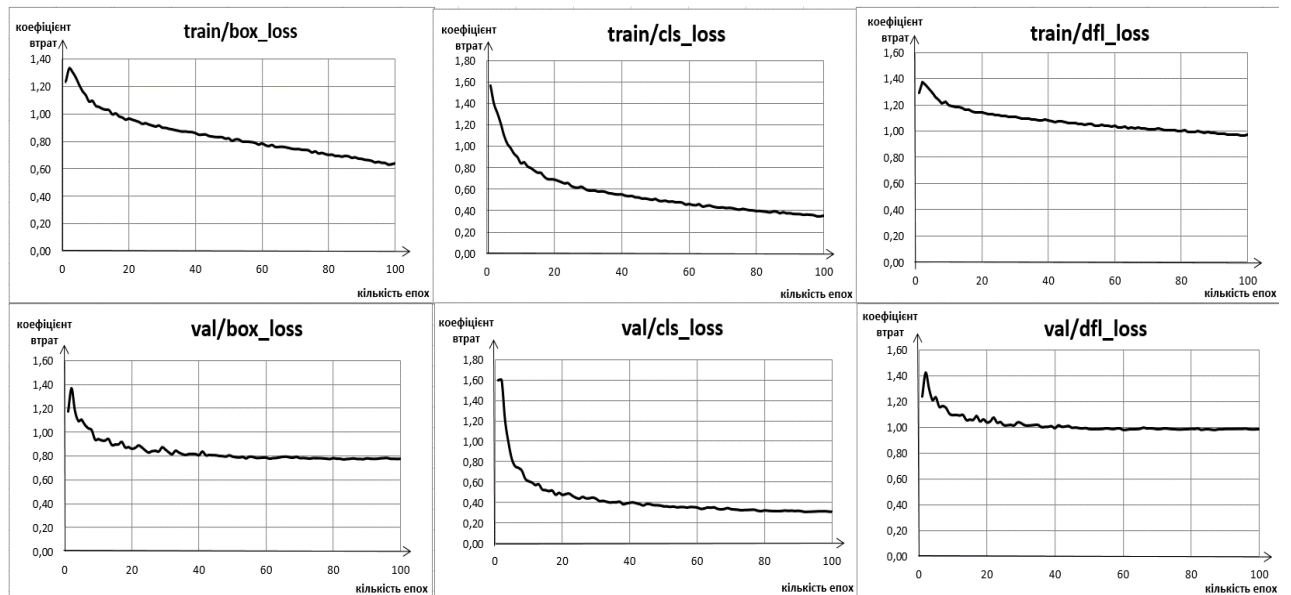


Рис. 2.12 – Графіки функції втрат

Детальне трактування кожного графіка.

Графік **"train/box_loss"** представляє зміну значень функції втрат для розташування об'єктів під час тренування моделі.

- Початкове значення функції втрат складає 1.2344.
- Значення функції втрат збільшується до 1.3307, а потім послідовно знижується впродовж перших 25 епох тренування.
- Після цього ми бачимо тренд зниження значень функції втрат з незначними варіаціями.
- Значення функції втрат досягає найнижчого значення 0.63148.

Закономірності, які можна помітити з цього графіка:

1. Початкові значення функції втрат можуть бути високими, але з часом вони знижуються, що свідчить про покращення точності моделі під час тренування.

2. Впродовж перших 25 епох ми спостерігаємо значні зміни в значеннях функції втрат, після чого варіації стають менш помітними.
3. Існує тренд зниження значень функції втрат протягом усього тренування, що свідчить про поліпшення точності моделі.
4. Найнижче значення функції втрат досягається на останній епосі, що може вказувати на оптимальне навчання моделі на даному датасеті.

Загалом, графік "train/box_loss" демонструє позитивну динаміку під час тренування моделі, зниження функції втрат і покращення точності розташування об'єктів.

Графік "train/cls_loss" також представляє зміну значень функції втрат, але цього разу для класифікації об'єктів під час тренування моделі.

Аналіз детальніше:

- Початкове значення функції втрат складає 1.5674.
- Значення функції втрат зменшується до 1.2186 протягом перших п'яти епох тренування.
- Після цього ми спостерігаємо зменшення значень функції втрат з незначними варіаціями.
- Значення функції втрат досягає найнижчого значення 0.34742.

Закономірності, які можна помітити з цього графіка:

1. Початкові значення функції втрат можуть бути високими, але з часом вони зменшуються, що свідчить про покращення точності класифікації моделі під час тренування.
2. Впродовж перших п'яти епох ми спостерігаємо значне зменшення значень функції втрат, після чого варіації стають менш помітними.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Існує тренд зниження значень функції втрат протягом усього тренування, що свідчить про поліпшення точності класифікації моделі.
4. Найнижче значення функції втрат досягається на останній епосі, що може вказувати на оптимальне навчання моделі на даному датасеті.

Загалом, графік "train/cls_loss" показує позитивну динаміку під час тренування моделі, зниження функції втрат і покращення точності класифікації об'єктів.

Графік "train/dfloss" представляє зміну значень функції втрат під час тренування моделі для виявлення областей зміни (деталів) на зображенні.

Аналіз детальніше:

- Початкове значення функції втрат складає 1.2891.
- Значення функції втрат показує певне зниження до 1.1964 протягом перших декількох епох тренування.
- Після цього ми спостерігаємо варіації значень функції втрат з незначним зростанням і зниженням.
- Значення функції втрат досягає найнижчого значення 0.96553.

Закономірності, які можна помітити з цього графіка:

1. Початкові значення функції втрат можуть бути високими, але з часом вони зменшуються, що свідчить про покращення точності виявлення зміни областей на зображенні під час тренування моделі.
2. Значення функції втрат показує невелике зниження протягом перших декількох епох, що свідчить про покращення виявлення деталей на зображенні.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Після цього спостерігається незначні варіації значень функції втрат, що може означати, що модель досягла певного стабільного рівня виявлення деталей.
4. Найнижче значення функції втрат досягається на останній епосі, що може вказувати на оптимальне навчання моделі на даному датасеті для виявлення зміни областей.

Загалом, графік "train/dfl_loss" вказує на покращення точності виявлення деталей на зображенні під час тренування моделі. Однак, варто відзначити, що більш детальний аналіз може бути зроблений лише з більшим обсягом даних та інформацією про саму модель та її параметри.

Аналізуючи графік "val/box_loss" детально, ми можемо зробити наступні висновки для моделі:

1. Початкові значення функції втрат високі і становлять близько 1.17-1.37.
2. Після приблизно 40 епох тренування, значення функції втрат починають знижуватися і досягають приблизно 0.77-0.88 після 20-30 епох.
3. Після досягнення цього рівня, значення функції втрат коливаються навколо певного значення без значних змін.
4. Значення функції втрат можуть знижуватися або зростати трохи протягом наступних епох, але загалом залишаються стабільними.

Загальна закономірність, яку можна виокремити з цього графіка, полягає в тому, що значення функції втрат для локалізації об'єктів стабілізуються на рівні приблизно 0.77-0.88 після приблизно 40 епох тренування.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

Враховуючи надані значення функції втрат **"val/cls_loss"** можна встановити наступні тенденції:

1. На початку тренування значення функції втрат високі (близько 1.2-1.6).
2. З кожною наступною епохою значення поступово зменшуються.
3. Після досягнення значення близько 0.3-0.4 (після близько 30-40 епох) відбуваються коливання в межах цього діапазону.
4. Загалом, значення функції втрат стабілізуються навколо рівня 0.3-0.4 після певної кількості епох тренування.

Висновки для значення функції втрат **"val/cls_loss"**:

- Тренування моделі успішне, оскільки значення функції втрат знижуються протягом початкових епох.
- Після досягнення певного рівня, функція втрат стабілізується, що може свідчити про досягнення плато в тренуванні моделі.
- Коливання значень функції втрат навколо певного рівня свідчать про наявність шуму або невеликих змін у тренувальних даних.
- Модель може потребувати додаткових оптимізаційних методів або архітектурних змін, якщо бажано подальше покращення результатів.

Аналізуючи графік **"val/dfloss"** детально, ми можемо зробити наступні висновки для моделі:

1. Початкові значення функції втрат для деформаційних зразків високі і становлять близько 1.2-1.4.
2. Протягом перших кількох епох тренування значення функції втрат знижуються, але не досягають стабільного рівня.
3. Після досягнення певної кількості епох, значення функції втрат починають коливатися навколо певного рівня без значних змін.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						61
Зм.	Арк.	№ докум.	Підпис	Дата		

4. Значення функції втрат можуть знижуватися або зростати трохи протягом наступних епох, але загалом залишаються стабільними.

Загальна закономірність, яку можна виокремити з цього графіка, полягає в тому, що значення функції втрат для деформаційних зразків стабілізуються на рівні близько 0.98-1.02 після певної кількості епох тренування.

Висновки для значення функції втрат " **val/dfl_loss** ":

- Тренування моделі для деформаційних зразків відбувається успішно, оскільки значення функції втрат знижуються протягом початкових епох.
- Значення функції втрат для деформаційних зразків стабілізуються навколо рівня 0.98-1.02 після певної кількості епох тренування.
- Коливання значень функції втрат навколо певного рівня свідчать про наявність шуму або невеликих змін у тренувальних даних.
- Модель може потребувати додаткових оптимізаційних методів або архітектурних змін, якщо бажано подальше покращення результатів.

Проаналізувавши якість мережі за метриками, отримуємо такі графіки, що зображенні на рис. 2.13.

metrics/precision(B): Графік відображає зміну точності (precision) моделі для виявлення об'єктів на тренувальному наборі даних. Точність вказує на частку правильно визначених об'єктів серед всіх виявлених об'єктів;

metrics/recall(B): Графік показує зміну показника повноти (recall) моделі для виявлення об'єктів на тренувальному наборі даних. Повнота вказує на частку правильно виявлених об'єктів серед усіх реальних об'єктів;

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

metrics/mAP50(B): Графік відображає зміну середньої точності під кривою Precision-Recall (mAP) при порозі 0.5 на тренувальному наборі даних. Це метрика, яка оцінює загальну якість виявлення об'єктів;

metrics/mAP50-95(B): Графік показує зміну середньої точності під кривою Precision-Recall (mAP) на тренувальному наборі даних при порогах від 0.5 до 0.95. Це дозволяє оцінити точність виявлення об'єктів на різних рівнях впевненості.

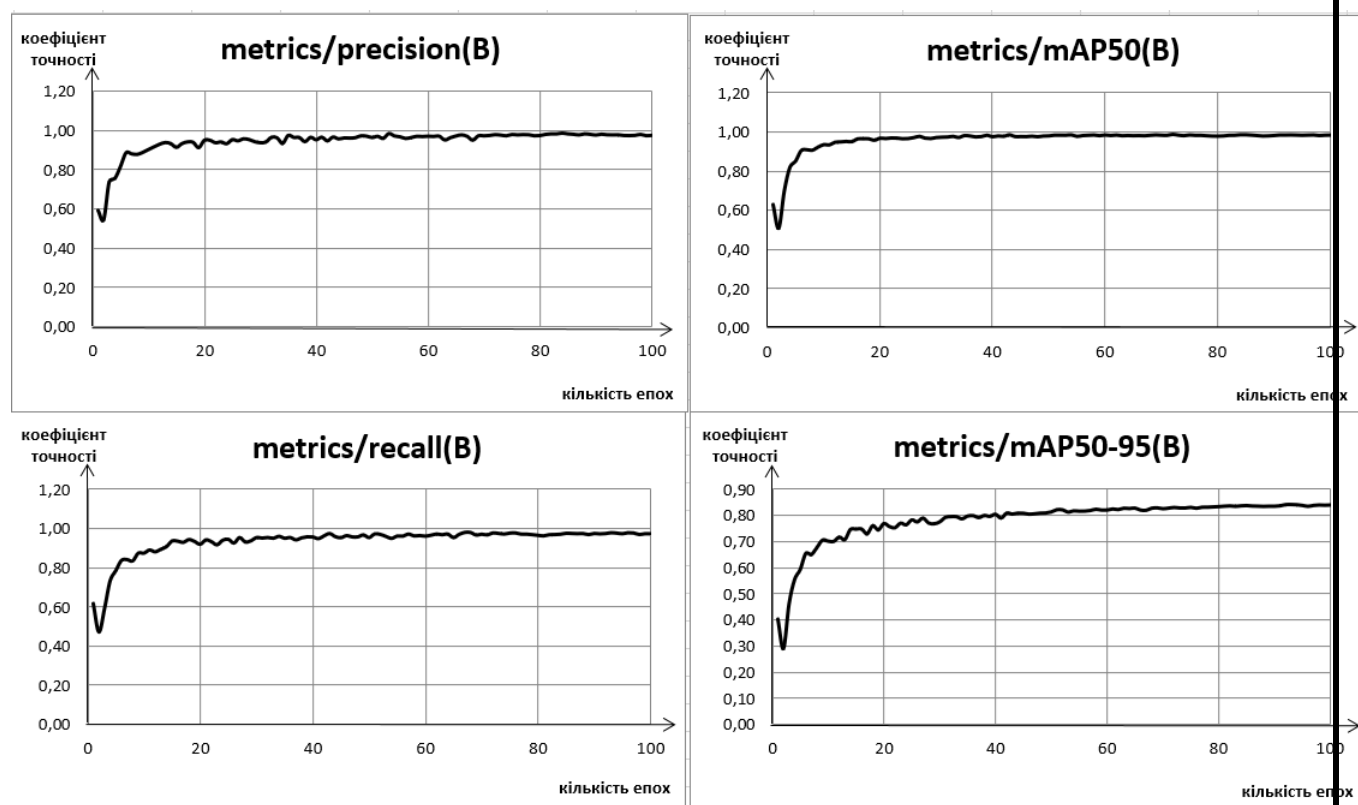


Рис. 2.13 - Метрики

Аналізуючи надані дані і графіки метрик, можна зробити наступні висновки:

1. Precision (точність). Значення метрики precision показує, наскільки точно модель класифікує позитивні приклади. За перші кілька епох точність

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

невелика, але з часом покращується і досягає значень від 0.95 до 0.98. Це свідчить про здатність моделі добре розрізняти позитивні та негативні приклади.

2. Recall (повнота). Recall вказує на здатність моделі знаходити всі позитивні приклади. Значення recall також покращуються протягом тренування і досягають значень від 0.95 до 0.98. Це означає, що модель здатна ефективно виявляти більшість позитивних прикладів у вхідних даних.

3. mAP50 (середній показник точності за порогом 0.5). mAP50 є середнім значенням точності для різних порогів відсікання (thresholds). Значення mAP50 зростають протягом тренування і досягають значень від 0.98 до 0.99. Це означає, що модель має високу середню точність у визначенні об'єктів з високим порогом відсікання.

4. mAP50-95 (середній показник точності в діапазоні від 0.5 до 0.95). mAP50-95 також показує середню точність, але у відрізку від 0.5 до 0.95. Значення mAP50-95 також покращуються протягом тренування і досягають значень від 0.8 до 0.85. Це свідчить про ефективність моделі в розпізнаванні об'єктів у широкому діапазоні порогів відсікання.

Загалом, з графіків видно, що модель поступово покращується протягом тренування, збільшуючи точність, повноту та середню точність. Це говорить про успішність тренування моделі в розпізнаванні об'єктів.

Попередній огляд роботи нейронної мережі з датасетом для перевірки демонструє, що нейронна мережа навчилася дуже добре, і що можна перевіряти її у більш складному завданні, що ми бачимо на рис. 2.14.

					<i>ДПБ.ПМ-91.05.1760.01.000.ПЗ</i>	Арк
						64
Зм.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.14 - Тестові зображення для перевірки нейронної мережі після тренування

Перевірка нейронної мережі

Для перевірки працездатності нейронної мережі достатньо, щоб вона обробила тестове відео, як потік фотографій, що реалізується наступним кодом, як на рис. 2.15.

```
[ ]
# Виконуємо прогнозування за допомогою моделі YOLO
!yolo detect predict model=my_weights.pt source="test_video_drone.mp4"

# Застосовуємо ffmpeg для обробки відео
ffmpeg -i {"runs/detect/predict/test_video_drone.mp4"} -vcodec libx264 {"final.mp4"}
```

Рис. 2.15 - Завантаження потоку зображення

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

Цей фрагмент коду встановлює бібліотеку Ultralytics, виконує прогнозування за допомогою моделі YOLO на вхідному відео "test_video_drone.mp4" та використовує ffmpeg для обробки результуючого відео та збереження його як "final.mp4". Коментарі над кожним рядком пояснюють його призначення.

Результат обробки потоку зображення показано на рис. 2.16.



Рис. 2.16 - Обробка потоку зображень нейронною мережею

Отже, потік зображень наша нейронна мережа доволі добре обробила і довела свою працездатність.

Наступним прикладом підключаємо свій потік зображень у реальному часі. Для цього потрібно завантажити ваги нейронної мережі та використати наступний алгоритм дій, описаний нижче у коді, зображений на рис. 2.17. Для цього використано середовище розробки pyCharm IDEA.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						66
Зм.	Арк.	№ докум.	Підпис	Дата		

```

predict_stream.py
1  # Імпортуємо необхідні бібліотеки
2  from ultralytics import YOLO
3  from ultralytics.yolo.v8.detect.predict import DetectionPredictor
4  import my_video_stream_connection
5  import cv2
6
7  # Ініціалізуємо модель YOLO з використанням вагів "my_weight"
8  model = YOLO("my_weight")
9
10 # Виконуємо прогнозування за допомогою моделі на вхідному відео зі з'єднанням my_video_stream_connection.stream,
11 # і відображаємо результати
12 results = model.predict(source=my_video_stream_connection.stream, show=True)
13

```

Рис. 2.17 Підключення безперервного потоку зображень до нейронної мережі

У цьому фрагменті коду імпортовано необхідні бібліотеки, ініціалізовано модель YOLO з використанням ваг "my_weight", і виконано прогнозування за допомогою цієї моделі на вхідному відео, яке отримано зі з'єднанням **my_video_stream_connection.stream**. Також, використовуючи параметр **show=True**, вказано моделі відображати результати прогнозування.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						67
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 2

В розділі 2 ми розглянули систему та складові компоненти нашої дипломної роботи. Ми проаналізували технічні та програмні компоненти, які були задіяні у процесі розробки та тренування моделі. Було розглянуто математичне моделювання навчання нейронної мережі та проведений огляд отриманих результатів.

За результатами тренування моделі для деформаційних зразків, ми спостерігали успішне зниження значень функції втрат протягом початкових епох. Значення функції втрат стабілізувалися навколо рівня 0.98-1.02 після певної кількості епох тренування. Коливання значень функції втрат свідчать про наявність шуму або невеликих змін у тренувальних даних. Для подальшого покращення результатів моделі можуть знадобитись додаткові оптимізаційні методи або архітектурні зміни.

В аналізі метрик, ми спостерігали покращення точності (precision) та повноти (recall) моделі протягом тренування. Значення точності досягали від 0.95 до 0.98, що свідчить про здатність моделі класифікувати позитивні приклади з високою точністю. Значення повноти також зростали і досягали подібних значень, вказуючи на здатність моделі виявляти більшість позитивних прикладів.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У цій дипломній роботі була розроблена автоматизована фотометрична система для розпізнавання об'єктів з використання мереж глибокого навчання. Завдяки використанню нейронних мереж досягається значне покращення у точності та ефективності системи, за рахунок тренування нейронної мережі для особистих задач.

Перевагою розробленої системи є її здатність до автоматизації процесу розпізнавання об'єктів. Операційна система, що використовується в розробленій системі, надає можливість навчання нейронної мережі, що дозволяє пристосовувати її до різних типів об'єктів та умов зйомки. Це дозволяє налаштовувати систему для конкретних потреб користувача та забезпечує гнучкість в роботі з системою.

Розроблена нейронна мережа має високу точність розпізнаванні об'єктів завдяки своїй глибокій архітектурі та ретельному навчанню на великому обсязі даних. Вона здатна виявляти об'єкти з високою точністю та швидкістю, що робить її ефективним інструментом для фотометричних досліджень та автоматизованої обробки зображень.

Також варто відзначити, що розроблена АФ система з нейронною мережею дозволяє знизити втручання людини в процес розпізнавання об'єктів. Із застосуванням навченої нейронної мережі та автоматичного аналізу зображень, значно зменшено ручну роботу та помилки, що можуть виникати при ручному розпізнаванні об'єктів. Це забезпечує високу швидкість та надійність в роботі системи.

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

- [1] www.kdbsecurity.nl [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.kdbsecurity.nl/camera-pakketten/855-hikvision-ip-camerabewaking-set-4-dome-camera-s-8-mp.html>
- [2] www.avicam-kameras.de [Електронний ресурс] – Режим доступу до ресурсу:
https://www.avicam-kameras.de/Dahua-2MP-außen-Überwachungskamera-Set-4-Turret-Kameras-Mikrofon-Nachtsicht-30-Meter?gclid=Cj0KCQjw7PCjBhDwARIsANo7CgmIhn1HQVAJD_NkuJyJQHZNRLAjaJKdpJPEFqrA4oBGkm6C-60yn-oaAht5EALw_wcB
- [3] Анатомія, фізіологія, еволюція нервової системи Ірина Маруненко, Євгенія Неведомська, Ганна Волковська
- [4] blog.goodaudience.com [Електронний ресурс] – Режим доступу до ресурсу:
<https://blog.goodaudience.com/comparison-of-the-brain-with-a-computer-neural-network-3ecd3c1fef57>
- [5] <http://www.rikas-varta.com.ua> [Електронний ресурс] – Режим доступу до ресурсу:
http://www.rikas-varta.com.ua/index.php?inx=prd.cam.vl&prt=VL-8739D#tab_Info
- [6] xmeye.com.ua [Електронний ресурс] – Режим доступу до ресурсу:
https://xmeye.com.ua/ua/kommutatory/1864-poe-kommutator-svitch-48v-10-portov-8poe2uplink.html?gclid=CjwKCAjwvpcKbHb4EiwAujULMqkjg9m3IkyasKIYyB1VtHbGu937FJTULKq1x9d9dcq5xU3WkuTmxoCbgkQAvD_BwE
- [7] Документація ultralytics [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.ultralytics.com/>
- [8] Документація YOLO (github) [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/ultralytics/ultralytics/tree/main/ultralytics/yolo/v8>

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

[9] Документація [keras](https://keras.io/) [Електронний ресурс] – Режим доступу до ресурсу:

<https://keras.io/>

[10] Документація tensorflow [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.tensorflow.org>

[11] Документація opencv [Електронний ресурс] – Режим доступу до ресурсу:

<https://opencv.org/>

Документація pytorch [Електронний ресурс] – Режим доступу до ресурсу:

<https://pytorch.org/>

[12] Онлайн застосунок CVAT (Computer Vision Annotation Tool) [Електронний ресурс] – Режим доступу до ресурсу: <https://app.cvat.ai>

[13] Данні для випробувань - kaggle [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset>

[14] Данні для випробувань - kaggle [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset>

[15] Датасет для навчання - [Електронний ресурс] – Режим доступу до ресурсу:

<https://universe.roboflow.com/project-ddrone/datasetdrone-trv98>

[16] «Прогнозування фінансових ринків методами машинного навчання»

Фоменко Нікіта Андріович – Режим доступу до джерела:

https://ela.kpi.ua/bitstream/123456789/40673/1/Fomenko_magistr.pdf

[17] «Інтелектуальний контент-аналіз соціальних мереж» - Цуканов Іван Володимирович

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

[18] machine-learning.paperspace.com [Електронний ресурс] – Режим доступу до ресурсу: <https://machine-learning.paperspace.com/wiki/convolutional-neural-network-cnn>

[19] Hough Transform [Електронний ресурс] – Режим доступу до ресурсу: <https://www.infoq.cn/article/3KfCkf67yZQHuiFp0x3Q>

					ДПБ.ПМ-91.05.1760.01.000.ПЗ	Арк
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

