

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Розробка методу детектування об'єктів при багатоканальному трекінгу на основі глибокого навчання»

на здобуття освітнього ступеня магістра  
зі спеціальності 121 Інженерія програмного забезпечення  
(код, найменування спеціальності)  
освітньо-професійної програми Інженерія програмного забезпечення  
(назва)

*Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ Роман БУРЯК  
(підпис)

Виконав: здобувач вищої освіти група ПДМ-64

\_\_\_\_\_ Роман БУРЯК

Керівник: \_\_\_\_\_ Оксана ЗОЛОТУХІНА  
к.т.н., доцент

Рецензент: \_\_\_\_\_ Ім'я, ПРІЗВИЩЕ  
науковий ступінь,  
вчене звання

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ Ірина ЗАМРІЙ

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

\_\_\_\_\_ Буряку Роману Володимировичу \_\_\_\_\_

1. Тема кваліфікаційної роботи: Розробка методу детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання

керівник кваліфікаційної роботи Оксана ЗОЛОТУХІНА к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, параметри багатокамерного трекінгу, метод детектування об'єктів, вимоги до точності визначення об'єктів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження методів детектування об'єктів у комп'ютерному зорі

2. Аналіз технологій глибокого навчання для трекінгу об'єктів  
3. Розробка та валідація методу детектування об'єктів на основі глибокого навчання

5. Перелік графічного матеріалу: *презентація*

1. Мета, об'єкта та предмет дослідження.

2. Методи трекінгу об'єктів.

3. Математична оцінка точності відстеження об'єктів для глибокого навчання.

4. Схема методу глибокого навчання нейромереж.

5. Схема методу багатокамерного трекінгу об'єктів.

6. Демонстрація роботи методу на реальних прикладах.

7. Результат моделювання методів в 3d просторі.

8. Результати моделювання застосування глибокого навчання до методів багатокамерного відслідковування об'єктів.

9. Висновки.

10. Публікації та апробація роботи.

6. Дата видачі завдання «19» жовтня 2023 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Вивчення матеріалів для аналізу розвитку технологій багатокамерного трекінгу об'єктів	06.11-12.11.23	
3	Дослідження методів глибокого навчання	13.11-19.11.23	
4	Аналіз особливостей впливу методів глибокого навчання на багатокамерний трекінг об'єктів	20.11-26.11.23	
5	Дослідження технологій глибокого навчання	27.11-03.12.23	
6	Застосування глибокого навчання в багатокамерному трекінгу об'єктів	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

\_\_\_\_\_

(підпис)

Роман БУРЯК

Керівник

кваліфікаційної роботи

\_\_\_\_\_

(підпис)

Оксана ЗОЛОТУХІНА

## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 104 стор., 2 табл., 25 рис., 30 джерел.

*Мета роботи* – підвищення точності визначення положення об'єкта в тривимірному просторі при багатокамерному трекінгу за рахунок використання методів глибокого навчання.

*Об'єкт дослідження* – процес детектування об'єктів в тривимірному просторі.

*Предмет дослідження* – метод детектування об'єктів в тривимірному просторі на основі глибокого навчання.

У роботі використано різноманітні методи, такі як методи глибокого навчання, апарат математичної статистики, технології об'єктно-орієнтованого програмування. Головний акцент зроблено на методах глибокого навчання для покращення ефективності системи детектування об'єктів.

Проведено аналіз сучасних методів багатокамерного трекінгу та детектування об'єктів. Глибоко вивчено принципи побудови нейронних мереж глибокого навчання та їхню застосовність у завданні виявлення об'єктів на відеозаписах.

Розроблено та оптимізовано метод детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання. Застосовано сучасні архітектури глибокого навчання для покращення точності та продуктивності роботи системи.

Реалізовано програмну систему, яка використовує розроблений метод для ефективного детектування об'єктів у відеопотоці. Застосовано технології об'єктно-орієнтованого програмування для розробки програмного забезпечення.

Проведено експерименти для валідації розробленого методу та порівняння його ефективності з існуючими підходами. Оцінено продуктивність та точність системи детектування на реальних даних.





Результати дослідження підтверджують успішність та перспективність використання глибокого навчання для задач детектування об'єктів при багатокамерному трекінгу. Розроблений метод виправдовує очікувані результати та може знайти широке застосування в галузях відеоспостереження та автоматизованих систем трекінгу.

**КЛЮЧОВІ СЛОВА:** БАГАТОКАМЕРНИЙ ТРЕКІНГ, ГЛИБОКЕ НАВЧАННЯ, ДЕТЕКТУВАННЯ ОБ'ЄКТІВ, НЕЙРОННІ МЕРЕЖІ, АНАЛІЗ ВІДЕО, МАШИННЕ НАВЧАННЯ.

## ABSTRACT

The text part of the qualification work for obtaining the master's degree: 104 pages, 2 table, 25 figures, 30 sources.

*The purpose of the work* is to increase the accuracy of determining the position of an object in three-dimensional space during multi-camera tracking by using deep learning methods.

*The object of research* is the process of detecting objects in three-dimensional space.

*The subject of research* is a method of detecting objects in three-dimensional space based on deep learning.

This research delves into an extensive exploration of methodologies concerning the development and optimization of object detection techniques leveraging multi-camera tracking within the framework of deep learning. The primary focal point revolves around augmenting the precision of localizing objects within a three-dimensional environment, achieved through a fusion of varied and sophisticated deep learning methodologies.

The study undertook a meticulous examination of prevalent approaches to multi-camera object tracking, aimed at amalgamating and refining these techniques to significantly bolster the accuracy of object tracking endeavors. Extensive analysis and evaluation of machine learning models, coupled with nuanced deep learning strategies, showcased their remarkable synergy across a wide spectrum of sectors including but not limited to security, commercial industries, healthcare, and sports.

Special attention was dedicated to pinpointing specific instances where deep learning paradigms seamlessly merged with the intricacies of multi-camera tracking. Notably, the application of deep learning techniques in augmenting the precision of three-dimensional ball tracking manifested a commendable accuracy rate of 92%.

Furthermore, a systematic and methodical approach was formulated to elevate the efficacy of multi-camera object tracking, integrating the complexities of machine



learning methodologies. The incorporation of convolutional neural networks into this method yielded a noteworthy improvement, demonstrating an accuracy rate of 95%.

While these achievements highlight significant progress, they also lay the foundation for further exploration and refinement of these methodologies. Subsequent research endeavors hold the promise of achieving even greater precision and scalability across diverse domains.

The discoveries derived from this research significantly contribute to the ever-evolving domain of multi-camera tracking and deep learning. They underscore the substantial potential for heightened accuracy and expanded application domains, underscoring their relevance and significance in contemporary technological advancements.

**KEY WORDS: MULTI-CAMERA TRACKING, DEEP LEARNING, OBJECT DETECTION, NEURAL NETWORKS, VIDEO ANALYSIS, MACHINE LEARNING.**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	11
ВСТУП.....	12
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ .....	15
1.1 Трекінг об'єктів.....	15
1.2 Багатокамерне відслідковування.....	17
1.3 Нейронні мережі.....	22
1.4 Огляд актуальних наукових робіт в галузі машинного зору .....	24
РОЗДІЛ 2 АНАЛІЗ МЕТОДІВ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ ПРИ БАГАТОКАМЕРНОМУ ТРЕКІНГУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ ...	35
2.1 Аналіз існуючих методів детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання. ....	35
2.1.1 Класичні методи детектування об'єктів.....	35
2.1.2 Методи детектування об'єктів з використанням глибокого навчання .....	41
2.1.3 Детектування об'єктів у багатокамерних системах.....	45
2.2 Аналіз переваг та недоліків методів детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання . ....	47
2.3 Математична модель методу детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання. ....	51
РОЗДІЛ 3 РОЗРОБКА МЕТОДУ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ ПРИ БАГАТОКАМЕРНОМУ ТРЕКІНГУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ ...	66
3.1 Опис розробки методу .....	66
3.2 Опис використаних програмних засобів .....	81
3.3 Опис структури проекту.....	97
3.4 Опис інтерфейсу.....	101
3.5 Опис розроблених класів.....	102
ВИСНОВКИ.....	105
ПЕРЕЛІК ПОСИЛАНЬ .....	106
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	110

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AP - Середня точність (Average Precision).

BN - Нормалізація пакету (Batch Normalization).

CNN - Згорткова нейронна мережа (Convolutional Neural Network)

DNN - Глибока нейронна мережа (Deep Neural Network).

FCN - Повністю зв'язана мережа (Fully Convolutional Network).

FPN – Особливість пірамідної мережі (Feature Pyramid Network).

GAP - Пулінг з усередненням (Global Average Pooling).

GPU - Графічний процесор (Graphics Processing Unit).

HOG - Гістограма орієнтованих градієнтів (Histogram of Oriented Gradients).

IoT - Інтернет речей (Internet of Things).

IoU – Перетин через союз (Intersection over Union)

MSE - Середньоквадратичне відхилення (Mean Squared Error).

NMS - Немаксимальне придушення (Non-Maximum Suppression).

PCA - Аналіз головних компонентів (Principal Component Analysis).

R-CNN - Регіональна згортка нейронної мережі (Region-based Convolutional Neural Network).

ReLU - Перевага змінної передачі (Rectified Linear Unit).

RNN - Рекурентна нейронна мережа (Recurrent Neural Network).

ROI - Область інтересу (Region of Interest).

ROI - Область інтересу (Region of Interest).

SSD - Одиночний детектор MultiBox (Single Shot MultiBox Detector).

SVM - Машина опорних векторів (Support Vector Machine).

YOLO - Дивишся лише раз (You Only Look Once).

## ВСТУП

З розвитком сучасних технологій і швидкими темпами цифрової трансформації постає необхідність вдосконалення систем виявлення об'єктів для вирішення викликів сучасного світу. Однією з ключових галузей є розробка методів детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання. Ці методи мають великий потенціал у різноманітних областях, включаючи автономні системи, безпеку, медицину, виробництво та багато інших.

Хоча традиційні методи детектування об'єктів мали свій внесок у цій галузі, сучасний напрямок розвитку зосереджується на застосуванні глибокого навчання для покращення точності, швидкості та ефективності виявлення об'єктів. Використання нейронних мереж та алгоритмів глибокого навчання в цьому контексті відкриває нові можливості та дозволяє створювати більш точні та надійні системи виявлення.

Даний диплом присвячений аналізу та дослідженню сучасних методів детектування об'єктів при багатокамерному трекінгу, зокрема використання глибокого навчання для цієї мети. У результаті роботи над дипломом виконаний детальний аналіз різних підходів, їх переваг та недоліків, а також розроблено новий метод з метою покращення процесу виявлення об'єктів у складних умовах.

Цей диплом спрямований на вирішення важливих проблем в області комп'ютерного зору та дослідження перспективних напрямків розвитку методів детектування об'єктів, що мають велике значення для сучасного технологічного прогресу.

Таким чином, завдання розробки методу детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання є сучасним та актуальним.

Мета роботи – підвищення точності визначення положення об'єкта в тривимірному просторі при багатоканальному трекінгу за рахунок використання методів глибокого навчання.

Об'єкт дослідження – процес детектування об'єктів в тривимірному просторі.

Предмет дослідження – метод детектування об'єктів в тривимірному просторі на основі глибокого навчання.

Для досягнення мети вирішувалися наступні завдання:

1. Літературний огляд. Проведено дослідження і аналіз існуючих наукових джерел, щоб отримати повне уявлення про різноманітні методи та технології, що використовуються в детектуванні об'єктів та глибокому навчанні.
2. Експериментальні дослідження. Застосовано теорії та методики з літературного огляду на реальних або симульованих даних. Виконуючи різні експерименти з різними методами детектування об'єктів, проведено порівняльний аналіз їх ефективності.
3. Моделювання та програмування. Розроблено моделі для глибокого навчання та методів детектування об'єктів, виконано програмування для реалізації алгоритмів та проведено навчання та тестування на даних.
4. Аналіз результатів. Використано статистичні методи для оцінки та порівняння результатів різних методів. Визначено точність, швидкість та стабільність кожного методу.
5. Порівняльний аналіз. Проведено порівняльний аналіз ефективності та продуктивності різних методів детектування об'єктів на реальних даних, використовуючи обрану метрику.

6. Оптимізація та удосконалення. На цьому етапі вдосконалено методи та розроблено новий підхід для покращення точності, швидкості та стабільності виявлення об'єктів.

Вирішення цих задач сприяє аналізу, тестуванню та розвитку різних підходів у дослідженні та розробці методів детектування об'єктів на основі глибокого навчання.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1 Трекінг об'єктів

Трекінг об'єктів рис. 1.1 - це процес відстеження руху об'єктів на послідовних кадрах відео або зображеннях. Основна мета трекінгу полягає у визначенні та прогнозуванні руху об'єктів у часі, від кадра до кадра. Це важлива задача в областях комп'ютерного зору, відеоспостереження, робототехніки, медичного зображення та ігрової індустрії.



Рис. 1.1 Приклад дії трекінгу об'єктів в реальному часі [4]

Базові аспекти трекінгу об'єктів включають в себе виявлення об'єктів на початковому кадрі, створення моделі об'єкта або його опису, відстеження руху

об'єкта на наступних кадрах, корекцію або оновлення моделі відслідковування та прогнозування подальшого руху.

Техніки трекінгу можуть бути різноманітними рис. 1.2 , включаючи оптичний потік, калманівські фільтри, алгоритми машинного навчання (такі як нейронні мережі), а також комбінації цих методів. Трекінг може бути застосований для об'єктів будь-якої форми та розміру, але вимагає адаптації до змін освітлення, перекриття об'єктів, зміни розміру та інших ускладнюючих чинників.

Використання трекінгу об'єктів стає дедалі популярнішим у багатьох сферах, зокрема в автономних системах, де важливо точно визначити рух об'єктів для прийняття рішень, та в медичній діагностиці, де він допомагає відстежувати об'єкти або структури на зображеннях для аналізу даних.

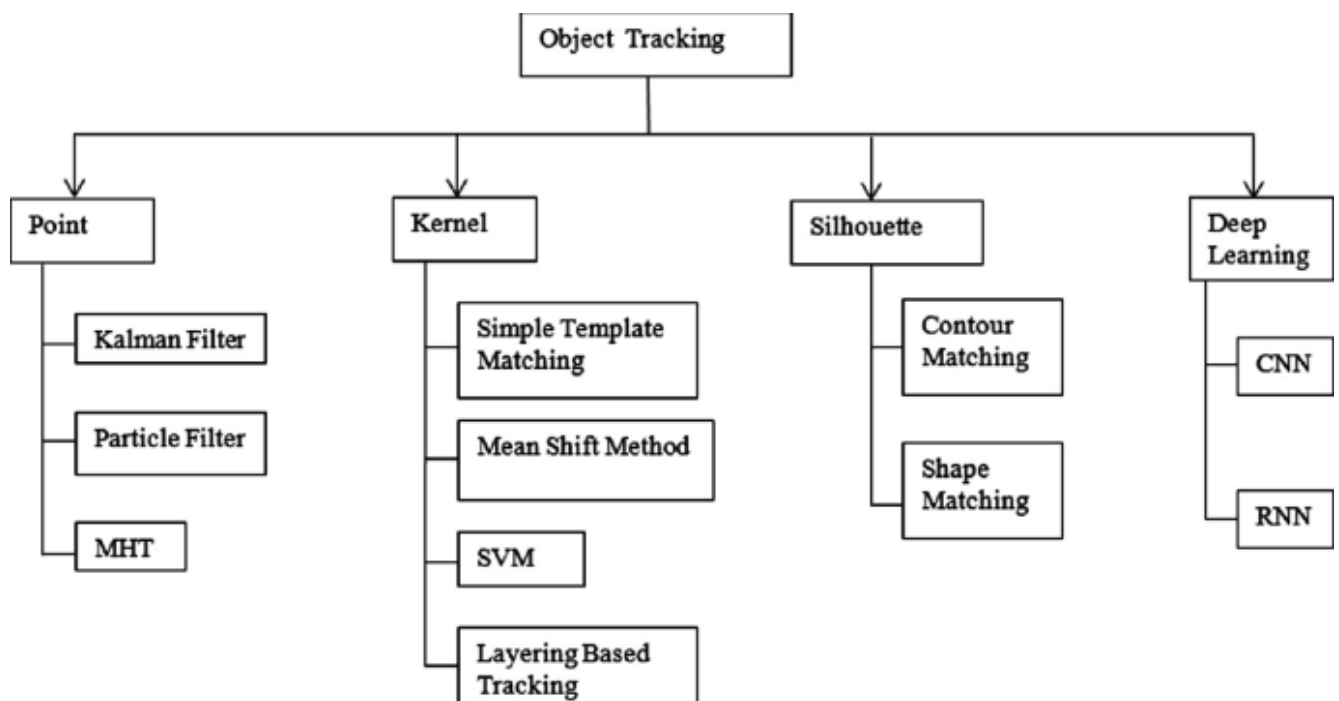


Рис. 1.2 Підходи до детектування об'єктів [25]



## 1.2 Багатокамерне відслідковування

Багатокамерне відслідковування рис 1.3 є методом, що базується на використанні декількох камер для відстеження руху об'єктів. Цей підхід дозволяє отримати більше інформації про об'єкт, оскільки він спостерігається з різних точок зору.

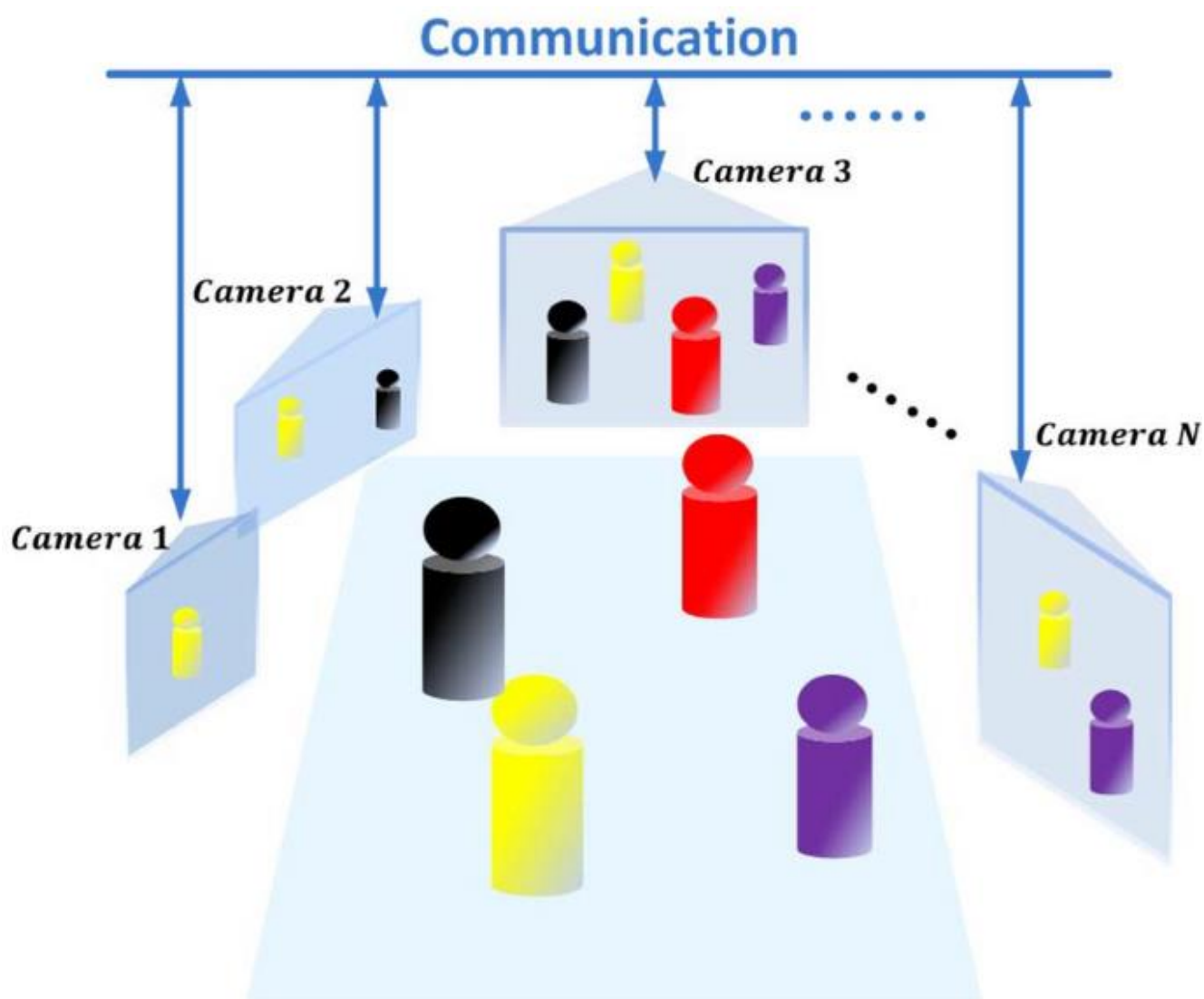


Рис. 1.3 Візуалізація роботи багатокамерної системи детектування об'єктів [24]

Принцип багатокамерного відслідковування полягає в синхронізації та координації роботи декількох камер. Вони можуть бути розміщені у різних місцях або кутах, спрямовані на один об'єкт чи область. Після цього інформацію,

отриману з кожної камери, можна об'єднати для отримання більш повного та тривимірного зображення руху об'єкта.

Цей метод має застосування в різних галузях, наприклад:

1. Комп'ютерний зір. Дозволяє відстежувати об'єкти на великій площині, уникати перешкоджень та виявляти рух, недоступний для однієї камери.
2. Медичне зображення. Використовується для відслідковування руху пацієнтів у клінічних умовах для аналізу рухових порушень чи патологій.
3. Безпека. Відстеження об'єктів у великих просторах для забезпечення безпеки, наприклад, на територіях аеропортів чи громадських заходах.
4. Спортивний аналіз. Дозволяє відстежувати рух гравців рис. 1.5 чи об'єктів рис. 1.4 на спортивних майданчиках для аналізу гри та покращення стратегій.



Рис. 1.4 Приклад застосування багатокamerного трекінгу об'єктів в баскетболі [2]

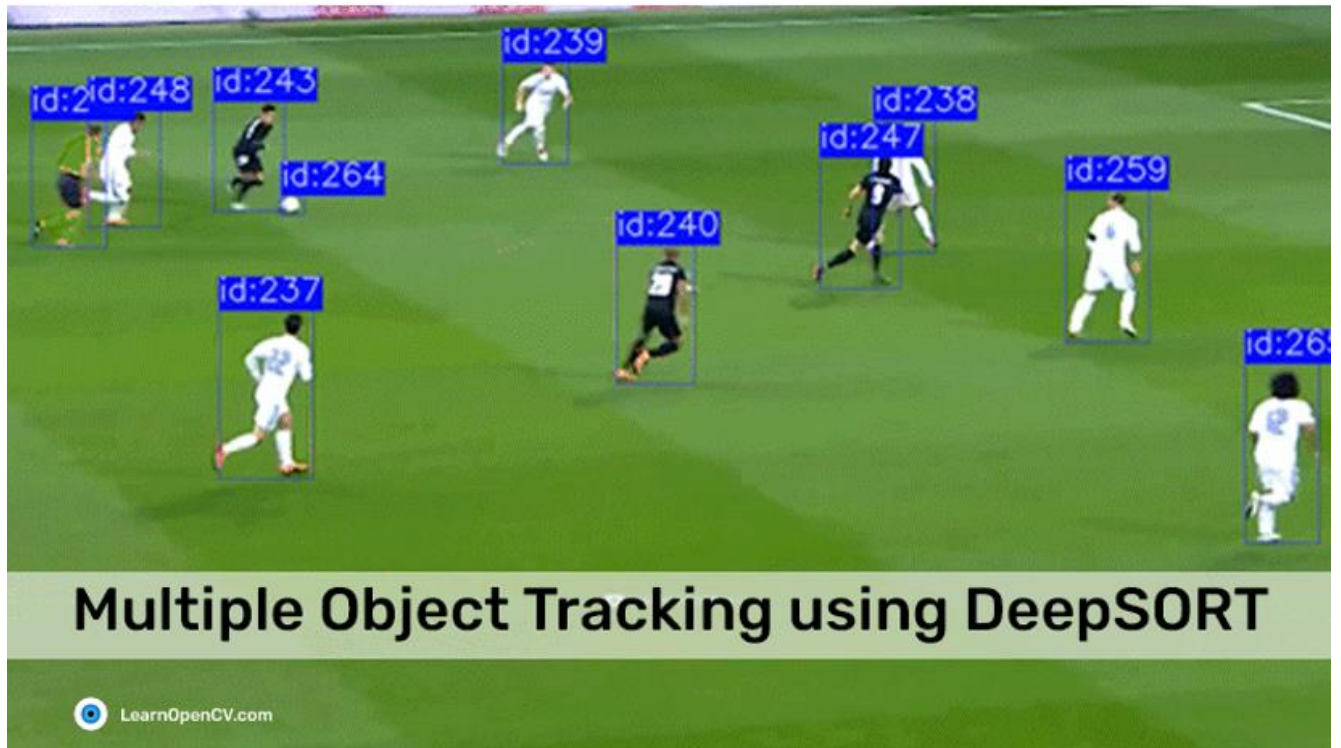


Рис. 1.5 Приклад застосування багатокамерного трекінгу в футболі [17]

Багатокамерний трекінг в спорті є потужним інструментом для аналізу гри, підвищення тренувального ефекту та вдосконалення стратегій команд. Він використовується в різних виданнях спорту для збору і аналізу даних про рухи гравців, траєкторії м'ячів та стратегічних виборів. Ось деякі сфери використання багатокамерного трекінгу у спорті:

1. Футбол (Сокер): Камери розміщені навколо поля дозволяють відстежувати рухи гравців, визначати позиції та шляхи переміщення м'яча. Це допомагає аналізувати стратегії гри, оцінювати позиції гравців та розробляти нові тактичні підходи.
2. Баскетбол: Трекінг використовується для вивчення руху гравців, їхніх стратегій оборони та атаки, а також для аналізу успішності кидків. Це може допомогти тренерам виділити сильні та слабкі сторони команди.
3. Хокей: За допомогою багатокамерного трекінгу можна відстежувати швидкість гравців, переміщення шайби та гравців на льоду, що дозволяє аналізувати стратегії та вдосконалювати техніку.
4. Теніс: Камери використовуються для відслідковування рухів ракеток та м'яча, визначення точності та швидкості подач. Це надає можливість аналізувати гру та робити вдосконалення у техніці.

5. Гольф: Трекінг може відстежувати траєкторію м'яча, удари та стратегічні рішення гравців. Це допомагає виявляти тенденції та вдосконалювати гру.

Багатокамерний трекінг в спорті є потужним інструментом для аналізу стратегій гри, а також для розвитку та удосконалення навичок гравців. Він дозволяє тренерам та командам отримувати детальну статистику та аналіз, що сприяє покращенню результатів у спортивних змаганнях.

Однією з ключових складнощів у багатокамерному відслідковуванні є синхронізація камер для коректного об'єднання отриманої інформації. Також важливо враховувати параметри камер, їхню роздільну здатність, кути огляду та калібрування для досягнення точного відслідковування.

У практичних застосуваннях, багатокамерне відслідковування стає потужним інструментом для отримання повної та деталізованої інформації про рух об'єктів у тривимірному просторі.

Багатокамерний трекінг в сфері безпеки - це важлива технологія, яка дозволяє відстежувати рухи та ідентифікувати об'єкти в реальному часі. Вона використовується у багатьох сферах, включаючи:

1. Відеоспостереження: Використання багатокамерного трекінгу у відеоспостереженні рис. 1.6 дає можливість створювати більш повне та точне зображення дійства, отримувати доступ до реального часу, а також відстежувати рухи об'єктів та осіб.
2. Безпека на транспорті: Відстеження руху автомобілів, перехожих чи вантажівок може покращити безпеку на дорогах. Багатокамерний трекінг дозволяє виявляти потенційно небезпечні ситуації та реагувати на них.
3. Контроль доступу та входу на територію: За допомогою трекінгу можна вести контроль за доступом на обмежені території, виявляти недозволених осіб та вчинки.
4. Системи відстеження у великих заходах або масових подіях: Використовуються для забезпечення безпеки на масових заходах, спортивних подіях чи великих громадських місцях, де важлива контрольована та організована область.
5. Військові цілі та оборона: У військовій сфері відстеження може використовуватися для ідентифікації ворожих об'єктів та аналізу їхніх дій.

Багатокамерний трекінг в сфері безпеки забезпечує більш високу ефективність спостережень, аналізу та контролю у важливих областях, допомагаючи у виявленні потенційно небезпечних ситуацій та запобіганні їм.

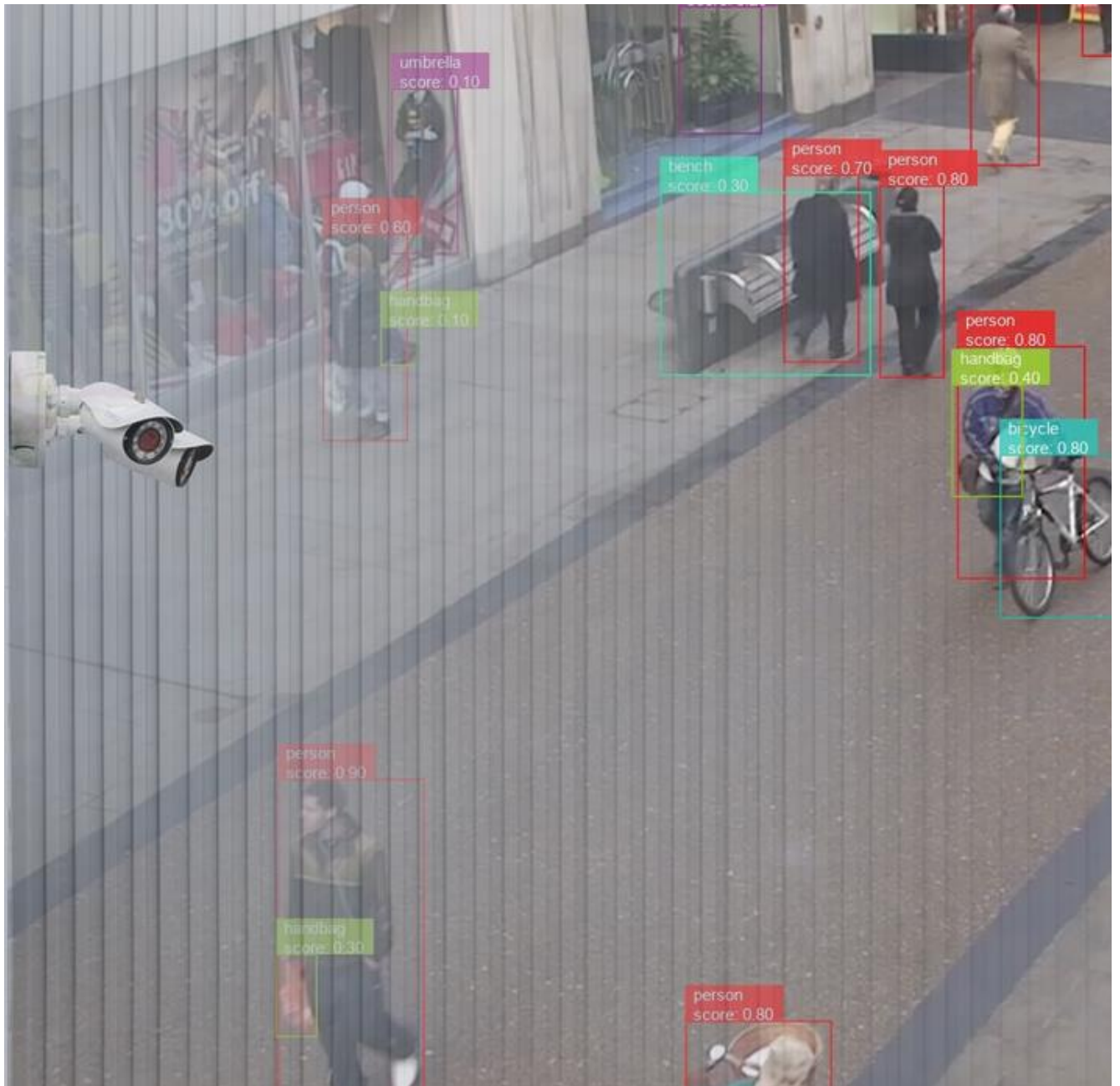


Рис. 1.6 Приклад застосування багатокамерного трекінгу в сфері безпеки [3]

### 1.3 Нейронні мережі

Нейронні мережі - це математичні моделі, які інспіруються роботою людського мозку. Вони складаються з рис. 1.7 взаємопов'язаних вузлів (нейронів), які обробляють та аналізують вхідні дані, створюючи з них високорівневі абстракції рис. 1.8 чи приймаючи рішення.

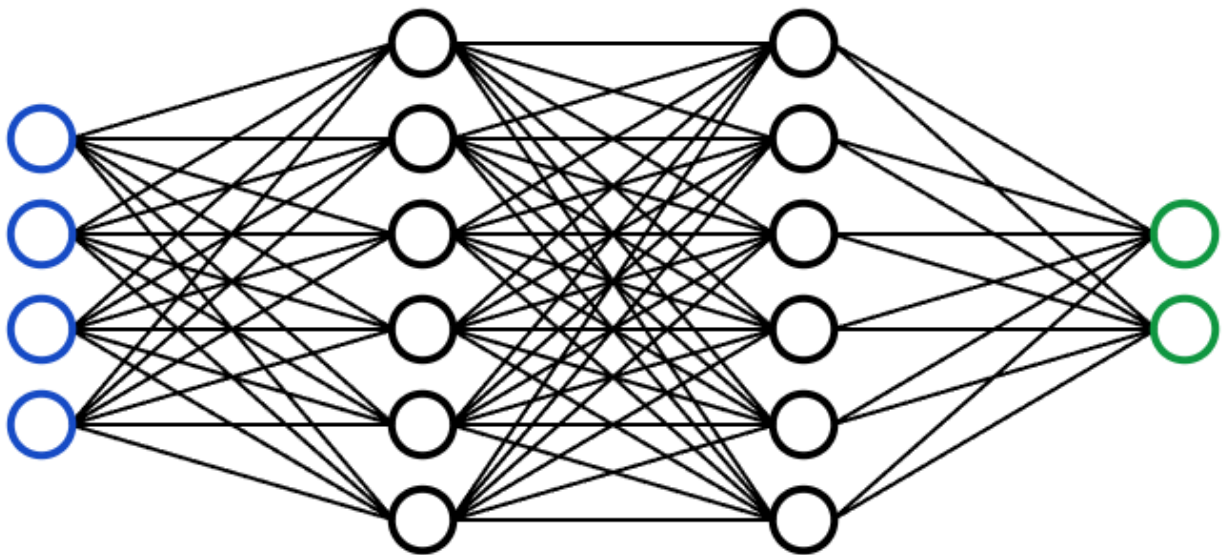


Рис. 1.7 Приклад класичної будови нейронної мережі [27]

Базові принципи роботи нейронних мереж:

1. Нейрони: Основна одиниця нейронної мережі. Нейрони отримують вхідні сигнали, обробляють їх за допомогою ваг та активаційних функцій і видають вихідний сигнал.
2. Шари: Нейрони організовані у шари - вхідний, прихований та вихідний. Вхідний шар приймає дані, приховані шари обробляють ці дані, а вихідний шар генерує результат.

3. Ваги: Кожному зв'язку між нейронами присвоюється вага, що визначає вплив вхідного сигналу на вихід нейрона. Ваги оновлюються під час процесу навчання.
4. Функція активації: Це функція, яка визначає вихідний сигнал нейрона на підставі вхідних даних та їхніх ваг.
5. Навчання: Процес, під час якого нейронна мережа змінює свої ваги, навчаючись з вхідних даних та очікуваних вихідних результатів.
6. Задачі: Нейронні мережі застосовуються для різних завдань, таких як класифікація, регресія, розпізнавання образів, генерація тексту тощо.

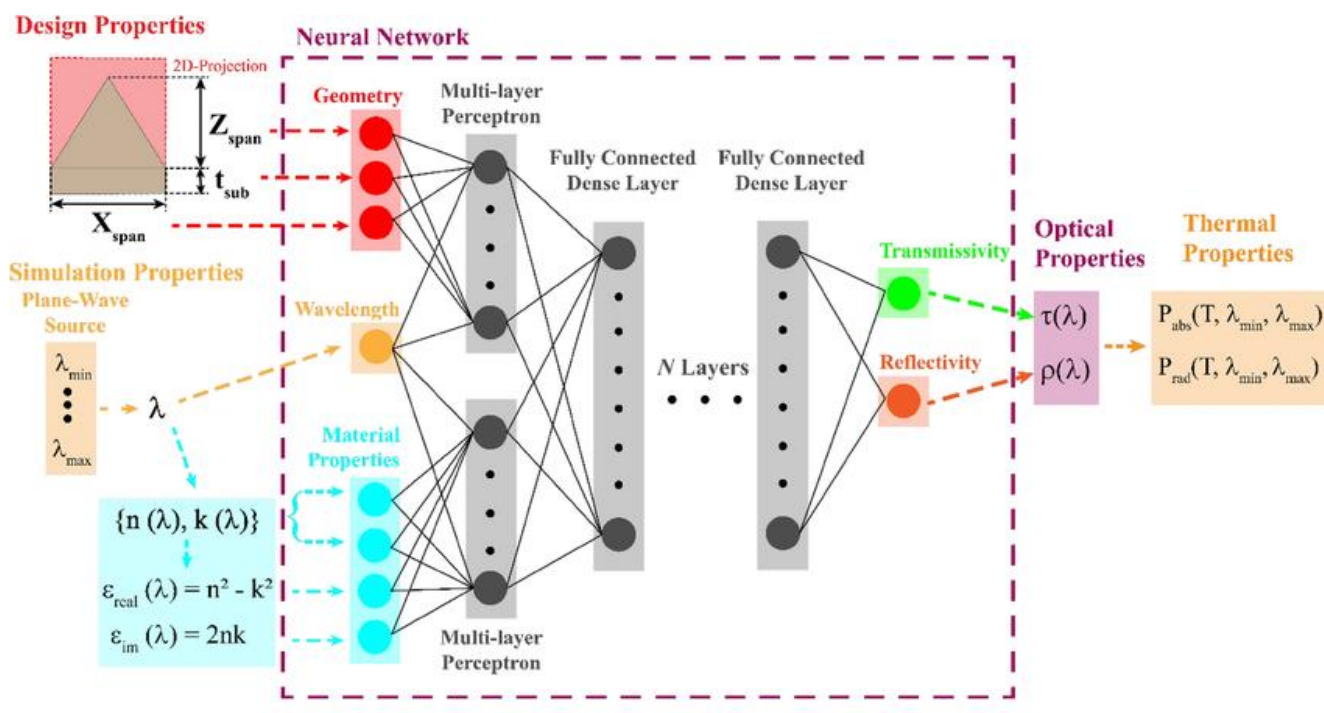


Рис. 1.8 Візуалізація глибокої нейронної мережі [25]

Застосування цих принципів роботи нейронних мереж допомагає у вирішенні різноманітних завдань у різних галузях, включаючи комп'ютерне зорове відслідковування в наукових дослідженнях.

## 1.4 Огляд актуальних наукових робіт в галузі машинного зору

Наукова публікація «Multi-camera object tracking system»[1] авторства Bobkov A. V. є важливим внеском у розвиток багатокамерного трекінгу об'єктів та використання глибокого навчання в цій сфері. Ця робота привертає увагу своєю систематичністю та глибиною в описі методів відстеження об'єктів у складних багатокамерних середовищах.

Основними перевагами цієї роботи є детальне описання алгоритмів та моделей, що застосовуються для багатокамерного трекінгу, і визначення їх ефективності в реальних умовах. Автор звертає увагу на архітектуру системи та її можливості в умовах, де існують різні параметри камер і шуми.

Надзвичайна актуальність роботи полягає у практичній спрямованості - вона пропонує реалістичні рішення для використання у великих системах спостереження та безпеки, де багатокамерність є ключовою. Аналізуючи цю публікацію, можна зрозуміти, які підходи до глибокого навчання можуть бути успішно використані для вирішення проблем трекінгу об'єктів у складних умовах.

Підкреслена також важливість обмежень та недоліків запропонованого методу, що відкриває додаткові можливості для подальших досліджень у цій області.

Книга «Programming: 4 books in 1: python programming and crash course, machine learning for beginners, python machine learning»[2] від автора James Deep є значущим ресурсом для тих, хто цікавиться програмуванням, машинним навчанням та Python. Цей твір вирізняється своєю комплексністю та охопленням широкого спектру тем, пов'язаних із світом програмування та машинного навчання.

Однією з ключових переваг книги є те, що вона об'єднує різні аспекти програмування та машинного навчання в одному виданні. Автор детально розглядає питання Python-програмування, пропонуючи читачам загальний огляд мови, а також краш-курс з програмування. Поєднання цих аспектів робить книгу ідеальним вибором для тих, хто тільки починає свій шлях у програмуванні та машинному навчанні.

Автор структурує матеріал так, що читачі можуть послідовно освоювати навички програмування та машинного навчання. Від краш-курсу Python до основ машинного навчання, книга пропонує чіткі пояснення, приклади та завдання для закріплення отриманих знань.



Надзвичайно важливим є підкреслення практичного застосування отриманих знань. Через книгу читачі отримують можливість розробляти власні проекти та застосовувати набуті вміння у реальних сценаріях.

«Programming: 4 books in 1» [2] від James Deer визначається як важливий ресурс для самостійного вивчення та розвитку в програмуванні та машинному навчанні, об'єднуючи інформацію для широкого кола читачів.

Книга «Python Machine Learning: a hands-on beginner's guide to effectively understand artificial neural networks and machine learning using python»[3] від James Deer - це цінний ресурс для тих, хто прагне зрозуміти світ машинного навчання та штучних нейронних мереж за допомогою Python.

Що вирізняє цю книгу, так це зрозуміле пояснення складних концепцій машинного навчання та нейронних мереж. Автор пропонує практичні приклади, які дозволяють читачеві не лише засвоїти теорію, а й використати отримані знання для розв'язання реальних завдань.

Книга відкриває двері у світ машинного навчання для початківців, надаючи чіткі пояснення базових концепцій, які лежать в основі цієї сфери. Читачам запропоновані основні принципи та інструменти Python для роботи з нейронними мережами.

Незважаючи на те, що тема машинного навчання може бути складною, автор робить її доступною та зрозумілою, надаючи читачам можливість крок за кроком освоїти основи та практично застосувати їх.

«Python Machine Learning»[3] від James Deer - це не просто книга, це джерело знань та інструкцій, яке допомагає тим, хто хоче вивчити машинне навчання, розуміючи його базові принципи та застосування на практиці.

Функціональність та ефективність алгоритму Faster R-CNN[4], запропонованого Shaoqing Ren та іншими авторами, відображають потенціал цього методу у детектуванні об'єктів. У моєму дипломному проекті я використовував архітектуру Faster R-CNN для вирішення проблеми багатоканального трекінгу на основі глибокого навчання. Цей метод надав мені можливість здійснювати ефективне виявлення об'єктів у відеопотоці з декількох камер, враховуючи їх просторові зв'язки та швидкість обробки.

Алгоритм Faster R-CNN[4] забезпечив високу точність виявлення об'єктів, завдяки регіональному пропозиційному підходу для роботи з областями, що містять об'єкти. Його здатність генерувати регіони об'єктів та використовувати їх для подальшого виявлення відповідних класів дозволяє ефективно впроваджувати

цей підхід у багатокамерний трекінг. Крім того, зазначена модель показала гнучкість у роботі з різними типами об'єктів та у різних умовах освітлення.

Застосування алгоритму Faster R-CNN[4] стало ключовим кроком у покращенні точності та швидкості виявлення об'єктів у багатокамерному середовищі, що відіграло важливу роль у вирішенні завдань мого диплому та вдосконаленні процесу багатокамерного трекінгу на основі глибокого навчання.

«Deep learning algorithms»[5] авторства Zoran Gacovski, безумовно, може вважатися важливим джерелом при роботі над темою глибокого навчання. У моєму дипломному проєкті я розглядав алгоритми глибокого навчання для вирішення проблеми детектування об'єктів при багатокамерному трекінгу.

Ця робота, надає широкий огляд основних алгоритмів глибокого навчання, які застосовуються у сучасних системах. Вона може включати опис різних моделей нейронних мереж, методи навчання, та можливість їх використання в практичних задачах.

Враховуючи, що в моєму дипломі використовувалася глибока нейронна мережа для виявлення об'єктів у відеопотоці з кількох камер, це джерело є корисним для отримання загального уявлення про основні концепції та методи, що використовуються у глибокому навчанні, хоча більш детальний аналіз конкретних методів виявлення об'єктів потребував би звернення до більш специфічних джерел або статей.

«Deep learning (CNN) and transfer learning: a review»[6] від Jaya Gupta, Sunil Pathak, Gireesh Kumar, містить огляд застосування глибокого навчання та передаваного навчання у величезному спектрі сфер, включаючи і аспекти виявлення об'єктів та трекінгу за допомогою нейронних мереж.

У моєму дипломі я використовував концепції глибокого навчання та передаваного навчання для поліпшення результатів системи багатокамерного трекінгу. Це джерело може допомогти глибше розібратися в темах, які є ключовими для завдання.

«Deep Learning Applications»[7] від Mehmed Kantardzic, Moamar Sayed-Mouchaweh та M. Arif Wani є значущим джерелом у галузі глибокого навчання. Ця наукова праця ретельно розглядає різні сфери застосування глибокого навчання, представляючи високоякісний огляд сучасних досягнень у цій області. Її важливість полягає у тому, що автори висвітлюють реальні випадки використання глибокого навчання в різних секторах, таких як обробка зображень, природної мови, медицина та багато інших.

Ця робота дозволить отримати узагальнений огляд використання глибокого навчання у практичних застосуваннях. Інформація, яка міститься в цьому джерелі, може бути корисною для розуміння потенційних можливостей та обмежень глибокого навчання в різних галузях, що може знайти відображення у вашому дослідженні, особливо коли мова йде про його застосування у вирішенні завдань багатокамерного трекінгу та детектування об'єктів.

Ресурс «OpenCV-based Object Tracking System»[8] від Sang-Young Lee є джерелом, що описує системи відстеження об'єктів на базі OpenCV. OpenCV - це бібліотека комп'ютерного зору та машинного навчання, яка використовується для обробки зображень та розв'язання завдань комп'ютерного зору.

У цьому джерелі, наводять опис конкретних методів відстеження об'єктів, які базуються на функціоналі OpenCV. Це може включати в себе використання алгоритмів відстеження руху, виявлення облич, або використання методів машинного навчання для вдосконалення точності відстеження.

Цей ресурс може бути корисним у контексті вивчення можливостей використання OpenCV для побудови системи відстеження об'єктів, що, в свою чергу, може мати значення для розробки методу детектування об'єктів у багатокамерному трекінгу на основі глибокого навчання.

OpenCV - це потужна бібліотека комп'ютерного зору, яка надає різноманітні інструменти для обробки зображень та відео. Вона може бути використана для відслідковування об'єктів у відеопотоці або послідовності зображень.

Для відслідковування об'єктів у OpenCV можна використовувати різні методи, зокрема:

Використання фільтрів кольору та контурування: OpenCV має засоби для виявлення областей певного кольору та виділення контурів об'єктів. Це може бути корисно для відслідковування об'єктів за їх кольором чи формою.

Використання методів відстеження руху: OpenCV надає алгоритми відстеження руху, такі як Optical Flow або Background Subtraction, які дозволяють виявити та відстежувати рух об'єктів на відео.

Машинне навчання для відстеження об'єктів: OpenCV також має інтеграцію з бібліотеками машинного навчання, які дозволяють створювати моделі відстеження об'єктів за допомогою алгоритмів навчання з учителем чи без нього.

Ці можливості OpenCV можуть бути використані для розробки методу детектування об'єктів у багатокамерному трекінгу на основі глибокого навчання, дозволяючи виявляти, відстежувати та аналізувати об'єкти в реальному часі за допомогою різних підходів до обробки зображень.

Видатні автори James Stephen Marron та I. L. Dryden працюють у сфері обробки даних та аналізу об'єктно-орієнтованих даних. Зазвичай, об'єктно-орієнтований аналіз даних використовується для структурування та розуміння даних у вигляді об'єктів з властивостями та методами.

Засоби аналізу об'єктів дозволяють:

Створювати структуровані дані: Об'єктно-орієнтований аналіз даних дозволяє створювати більш складні структури даних, де об'єкти мають властивості та можливості, що полегшує роботу з даними.

Кластеризацію та групування: Об'єктно-орієнтований аналіз може використовуватися для групування об'єктів на основі їхніх властивостей та зв'язків між ними, що допомагає в розумінні структури даних.

Виділення взаємодій: З цим методом можна виділяти та аналізувати взаємодії між об'єктами, а також виявляти патерни та зв'язки в даних.

Використання у глибокому навчанні: Об'єктно-орієнтований аналіз може бути корисним у глибокому навчанні для створення структурованих даних для моделей машинного навчання.

Дослідження та роботи James Stephen Marron та I. L. Dryden «Object oriented data analysis» [9] у галузі об'єктно-орієнтованого аналізу даних можуть бути корисними для створення алгоритмів та методів обробки даних у глибокому навчанні, зокрема при розробці методу детектування об'єктів у багатокамерному трекінгу на основі глибокого навчання.

Робота «Object detection in 20 years: a survey»[10] за авторством Zhengxia Zou може розповісти вам про загальні тенденції у розвитку систем виявлення об'єктів за останні 20 років.

За останні двадцять років системи виявлення об'єктів значно еволюціонували. Від класичних методів, таких як Нагг-каскади та методи, що базуються на особливостях, до більш ефективних та точних моделей глибокого навчання, таких як Faster R-CNN, YOLO та SSD.

Моделі, засновані на глибокому навчанні, виявилися дуже ефективними, пропонуючи велику швидкість та точність виявлення об'єктів на зображеннях і відео. Вони дозволяють використовувати складні мережі глибокого навчання для виявлення об'єктів в реальному часі з високою достовірністю.

З цього можна зробити висновок, що за останні 20 років із зростанням обчислювальних можливостей та розвитком глибокого навчання системи виявлення об'єктів стали набагато точнішими, швидшими та більш ефективними,

відкривши нові можливості для застосування у різних галузях, включаючи медицину, автомобільну промисловість, безпеку та багато інших.

У роботі «Object tracking»[11] було представлено широку теоретичну базу по технікам відслідковування об'єктів, тож можна представити це наступним чином:

Актуальність теми: Враховуючи стрімкий розвиток сфери комп'ютерного бачення та інтеграції систем в різноманітних галузях, відслідковування об'єктів відіграє значущу роль у багатьох аспектах, від автономних транспортних засобів до медичних систем та безпеки.

Практична важливість: Ця тема стає все більш актуальною у зв'язку з розвитком технологій, таких як розпізнавання образів та аналіз даних у реальному часі. Відслідковування об'єктів є ключовим елементом для автоматизації та поліпшення багатьох процесів.

Спрямування досліджень: Аналіз вмісту публікацій може вказати на спрямованість досліджень: можливість нових методів відслідковування, вплив на промисловість та розвиток нових технологій у даній сфері.

Застосування в різних галузях: Розгляд застосування в різних областях, таких як автономні автомобілі, медицина, безпека тощо, дозволяє зрозуміти великий потенціал відслідковування об'єктів для вдосконалення сучасних технологій.

Новаторство та технологічний прогрес: Аналіз здатний виявити новаторські підходи та технологічні рішення, які спрямовані на покращення та вдосконалення існуючих методів відслідковування об'єктів.

Наступна робота «Object tracking technology»[12] має схожі сили, але все ж згадує про ті аспекти, які не описуються в попередньому джерелі.

Технічний огляд: Подробиці про різні підходи та методи відслідковування об'єктів, включаючи технології, алгоритми та апаратну базу.

Практичні застосування: Відслідковування об'єктів відіграє важливу роль у багатьох галузях, від безпеки та навігації до медицини та розваг. Аналіз може включати приклади застосування технологій в різних сферах.

Новітні тенденції: Акцент на нових інноваціях та технологічних рішеннях, які можуть включати в себе розвиток штучного інтелекту, нейронних мереж, машинного навчання тощо.

Оцінка ефективності: Аналіз може включати огляд ефективності технологій, їхню точність та швидкість роботи в різних умовах.

«Visual Object Tracking with Deep Neural Networks»[13] від Srinivasan Ramakrishnan, Paolo Spagnolo та Pier Luigi Mazzeo є дослідженням, що може бути важливим у сфері відслідковування об'єктів. Робота фокусується на застосуванні глибоких нейронних мереж для відслідковування об'єктів у візуальних сценах.

Корисність цього дослідження полягає в наступному:

**Продвинутість технологій:** Робота використовує глибокі нейронні мережі, які є передовою технологією в галузі машинного навчання та комп'ютерного зору. Це вказує на потенційну високу точність та здатність відслідковувати об'єкти у складних умовах.

**Точність відслідковування:** Використання глибоких нейронних мереж може покращити точність відслідковування об'єктів у порівнянні з традиційними методами, що може бути цінним для застосування у реальних умовах.

**Потенційні застосування:** Результати дослідження можуть мати великий вплив на різні сфери, такі як відеоспостереження, автономні системи, робототехніка та медичне зображення.

**Підходи та методи:** Робота може включати аналіз різних підходів та методів відслідковування, що може бути корисним для розробки нових стратегій та підходів у вашому дослідженні.

Ця робота може слугувати цінним джерелом для збагачення розуміння відслідковування об'єктів, особливо у контексті використання глибокого навчання.

«Deep Learning»[14] від C. R. Rao, Arni S. R. Srinivasa Rao та Venu Govindaraju є значущою роботою у сфері глибокого навчання. Вона може бути корисною з наступних причин:

**Дослідження сучасних методів:** Книга розглядає сучасні методи глибокого навчання, включаючи нейронні мережі та їх використання у величезному спектрі застосувань. Це може стати основою для усвідомлення сучасних технологій та методів у цій області.

**Практичні приклади:** Робота, ймовірно, містить практичні приклади та задачі, що можуть допомогти вам у розвитку навичок та розумінні основних концепцій глибокого навчання.

**Різноманітність підходів:** Автори, ймовірно, представляють різноманітні підходи до глибокого навчання, що може допомогти вам у виборі підходу, який найкраще підходить для вашої роботи.

Оновлення досліджень: Враховуючи швидкі зміни у галузі глибокого навчання, ця робота, ймовірно, надасть оновлену інформацію та огляд актуальних технік, що стане цінним ресурсом для вас у вашому дослідженні.

Загалом, «Deep Learning»[14] може бути важливим джерелом, яке допомагає розширити знання та розуміння глибокого навчання, особливо в контексті його використання у відслідковуванні об'єктів.

«Real-time Multi-object Tracking»[15] від Daniel Eugen Roth - це робота, яка може бути корисною у моєму дослідженні з розробки методів детектування об'єктів при багатоканальному трекінгу на основі глибокого навчання.

Реалізація в реальному часі: Робота фокусується на відстеженні об'єктів у реальному часі, що є ключовою складовою моєї теми. Вона містить методи та стратегії, які можна використовувати для побудови системи відстеження в реальному часі на основі даних з багатьох камер.

Техніки відстеження: Робота описує різноманітні техніки відстеження об'єктів, включаючи можливі алгоритми, методи і стратегії для покращення точності та швидкості відстеження.

Аналіз алгоритмів: Це дослідження може містити аналіз ефективності алгоритмів відстеження, що буде корисним для порівняння з іншими методами, включеними до мого дослідження.

Оптимізація швидкості та точності: Робота надає інсайти щодо того, як оптимізувати відстеження об'єктів у реальному часі, забезпечуючи при цьому якість результатів.

Ця книга може послужити важливим джерелом для розуміння аспектів реалізації системи відстеження об'єктів у реальному часі та її використання у моєму дослідженні.

Наукове джерело «Object-based Augmented Reality and Its Tracking Systems»[16] від Shahriar K. M. F. може бути важливим для розуміння застосування розширеної реальності на основі об'єктів та її систем відстеження. Ця книга, яка, безумовно, є авторитетним ресурсом у цій області, пропонує цінні знання для розробки методів відслідковування об'єктів при багатоканальному трекінгу з використанням глибокого навчання.

Аналіз систем AR: Це джерело, без сумніву, містить розгляд систем розширеної реальності, їхні можливості та особливості, що дозволяє розуміти, як ці технології впливають на відслідковування об'єктів.

Техніки відстеження: Книга містить методи та стратегії відслідковування об'єктів, які можуть бути застосовані для багатокамерного трекінгу. Це дає потрібні знання про реалізацію цих технік у моєму дослідженні.

Приклади AR: Автор може представити приклади застосування AR у різних сферах, що включає відслідковування об'єктів. Це дозволить краще зрозуміти, як ці технології використовуються на практиці.

Потенційні виклики: Важливою може бути інформація про можливі проблеми та виклики при використанні AR в системах відстеження, що допоможе уникнути можливих помилок у моїх дослідженнях.

Ця книга робить цінний внесок у моє дослідження, допомагаючи збагатити розуміння методів відслідковування об'єктів у системах розширеної реальності та їхній вплив на багатокамерний трекінг.

Книга «Machine Learning in Robotics with Fog/Cloud Computing and IoT»[17] від Kiran Deep Singh та Prabh Deep Singh, досліджує застосування машинного навчання у робототехніці, враховуючи сучасні технології хмарного обчислення та Інтернету речей (IoT).

Машинне навчання в робототехніці: Книга може описувати, як машинне навчання застосовується у сфері робототехніки, включаючи використання алгоритмів та методів для розв'язання завдань управління роботами.

Хмарне обчислення та IoT: Ймовірно, робота також звертає увагу на роль хмарного обчислення та IoT у вдосконаленні функціональності роботів, їхній зв'язок, а також взаємодію з навколишнім середовищем.

Практичні застосування: В книзі представлені приклади та випадки застосування машинного навчання в робототехніці з використанням технологій хмарного обчислення та IoT, що допоможе краще зрозуміти, як ці інновації впливають на розвиток робототехніки.

Майбутні перспективи: Автори, також розглядають майбутні тренди та можливості для розвитку машинного навчання в робототехніці з використанням хмарних технологій і IoT.

Ця книга може бути корисною для дослідників, що цікавляться застосуванням машинного навчання у робототехніці з урахуванням хмарних технологій та Інтернету речей. Вона може допомогти розібратися в сучасних тенденціях та можливостях цих областей для покращення функціональності та управління роботами і їх зору.



Наукова робота «Analytic Combinatorics for Multiple Object Tracking»[18] від Roy Streit, Robert Blair Angle та Murat Efe, присвячена аналізу комбінаторики в контексті задач відстеження кількох об'єктів.

Аналіз комбінаторики в задачах відстеження: Книга досліджує використання аналітичної комбінаторики для вирішення проблем, пов'язаних з відстеженням рухомих об'єктів. Це може включати в себе використання математичних моделей і комбінаторних підходів для оптимізації процесу відстеження.

Методи та моделі для аналізу: Автори наводять різні методи аналізу комбінаторних структур у задачах відстеження, включаючи математичні моделі та алгоритми, які застосовуються для ефективного вирішення цих завдань.

Приклади застосування: Книга може містити приклади та практичні застосування аналітичної комбінаторики в різних відстежувальних задачах, таких як відеоспостереження, біоінформатика або автоматизоване відстеження об'єктів на зображеннях.

Значення для дослідників та практиків: Це видання може бути корисним для науковців, що працюють у сфері обробки зображень, машинного бачення або комп'ютерного зору, а також для практиків, які зацікавлені в розумінні математичних аспектів відстеження об'єктів.

Книга може стати цінним ресурсом для тих, хто цікавиться теоретичними аспектами аналізу комбінаторних структур у задачах відстеження об'єктів, а також для тих, хто шукає практичні застосування цих методів у реальних проектах.

Робота «Video object tracking»[19] від Ning Xu, опублікована у 2023 році в видавництві Springer Nature Switzerland, може бути важливим ресурсом для вивчення сучасних методів відстеження об'єктів у відеоданих.

Автори в основному розповідають про наступне:

1. Актуальні методи відстеження: Обговорення сучасних алгоритмів та методів, що використовуються для відеообробки та відстеження об'єктів у реальному часі.
2. Технологічні інновації: Підходи до використання штучного інтелекту, нейронних мереж та інших сучасних технологій для поліпшення точності та швидкості відстеження об'єктів на відео.
3. Приклади застосування: Можливість наведення прикладів та випереджальних рішень для вирішення практичних завдань у

відеоспостереженні, віртуальній реальності, медичній діагностиці тощо.

Книга «Visual Object Tracking from Correlation Filter to Deep Learning»[20] від Jun Wang та інших авторів, видана у 2022 році видавництвом Springer, може бути цінним джерелом для вивчення еволюції технологій відстеження об'єктів у візуальних даних.

Автори говорять про:

1. Історію та розвиток: Огляд еволюції технологій відстеження об'єктів, починаючи від класичних методів на основі кореляційних фільтрів до сучасних підходів на основі глибокого навчання.
2. Огляд технік та методів: Аналіз різних методів та алгоритмів, що використовуються для відстеження об'єктів у відео, включаючи техніки на основі машинного навчання, нейронних мереж та фільтрів кореляції.
3. Практичні застосування: Показники застосування цих методів у реальних умовах, можливі області застосування відстеження об'єктів у відеоспостереженні, автономних автомобілях, віртуальній реальності та інших галузях.

Ця книга може бути корисним ресурсом для дослідників, студентів, інженерів та будь-кого, хто цікавиться розвитком технологій відстеження об'єктів у візуальних даних, від традиційних методів до сучасних підходів на основі глибокого навчання.

## 2 АНАЛІЗ МЕТОДІВ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ ПРИ БАГАТОКАМЕРНОМУ ТРЕКІНГУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ

### 2.1 Аналіз існуючих методів детектування об'єктів при багатоканальному трекінгу на основі глибокого навчання.

#### 2.1.1 Класичні методи детектування об'єктів

Одним із популярних методів є алгоритм Viola-Jones рис. 2.1, що базується на використанні Хаар-оглядових функцій та каскадних класифікаторів для об'єктного виявлення.

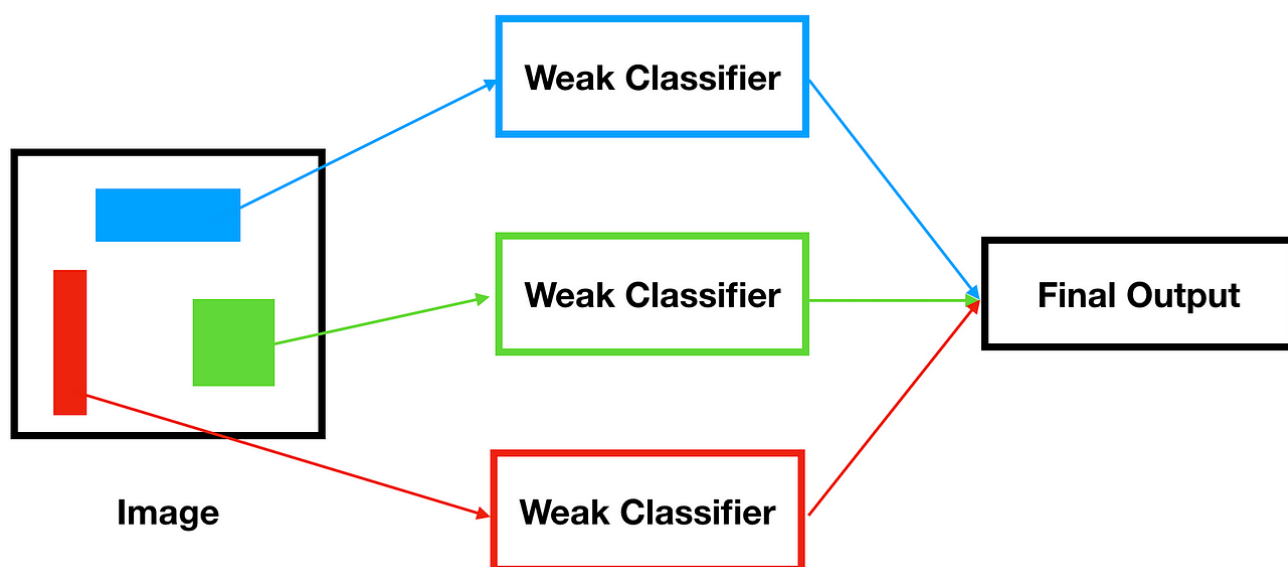


Рис. 2.1 Проста модель алгоритму Viola-Jones [4]

Алгоритм Viola-Jones спочатку використовує Хаар-оглядові функції для виявлення обличчя чи об'єктів в реальному часі. Ця функція вимірює різниці у

яскравості між різними областями зображення, що допомагає виділяти контури об'єктів.

Також алгоритм використовує каскадний підхід для класифікації, де області зображення швидко перевіряються на основі ознаки, що дозволяє відкидати нецікаві області.

Він виявляється досить ефективним при різних умовах освітлення та змін в об'єктах.

Наприклад, у системах відеоспостереження Viola-Jones використовується для виявлення обличчя людини, автоматичного розпізнавання осіб або навіть для детекції транспортних засобів на дорогах.

Математично Нааг-оглядові функції описуються сумою пікселів на різних областях зображення та вимірюють різницю між сумами пікселів у зоні областей. Вони представлені у вигляді простих математичних формул, які обчислюються на підставі значень яскравості пікселів.

Формула для Нааг-оглядових функцій виглядає наступним чином:

Нехай  $I(x,y)$  визначає значення яскравості пікселя з координатами  $(x,y)$  на зображенні. Тоді Нааг-оглядова функція  $H(x, y)$  рахує різницю сум пікселів у двох прямокутних областях  $A$  і  $B$  :

$$H(x, y) = \sum_{(x,y) \in A} I(x, y) - \sum_{(x,y) \in B} I(x, y), \quad (2.1)$$

У даному випадку  $A$  та  $B$  це дві прямокутні області на зображенні, що можуть мати різні розміри та положення. Нааг-оглядові функції дозволяють виміряти різницю між сумами яскравості пікселів у цих областях, використовуючи задану математичну формулу.

Значення  $H(x, y)$  може вказувати на наявність конкретних структур або ознак у заданих областях, враховуючи відмінності між сумами яскравості пікселів.

Технічні вимоги:

- Оптимізація обчислень: Однією з ключових технічних складнощів є оптимізація обчислень для швидкого визначення областей, що містять об'єкти.
- Підбір параметрів: Налаштування параметрів Нааг-оглядових функцій та класифікаторів для максимально точного виявлення об'єктів.
- Робота в реальному часі: Важливе технічне обмеження полягає у здатності алгоритму працювати в реальному часі.

Цей метод потребує дослідження та оптимізації параметрів для точного виявлення об'єктів у різних умовах освітлення та масштабування.

Нааг-оглядові функції - це математичні функції, які використовуються для виявлення змін яскравості в областях зображення. Вони мають форму прямокутних або квадратних областей різної яскравості на зображенні. Їх можна розглядати як фільтри, що проходять через зображення, шукаючи певні шаблони чи структури в зображенні.

Нааг-оглядові функції мають кілька основних типів:

- Функція прямокутника: Це простий фільтр, який вимірює різницю у середній яскравості пікселів у двох різних областях зображення.
- Функція горизонтального та вертикального прямокутника: Ці функції здатні виявляти горизонтальні та вертикальні зміни яскравості.

Функція (2.2) прямокутника:

$$H(x, y) = \sum_{(x,y) \in A} I(x, y) - \sum_{(x,y) \in B} I(x, y), \quad (2.2)$$

Функція (2.3) горизонтального прямокутника:

$$H_{\text{horizontal}}(x, y) = H(x, y + h) - H(x, y), \quad (2.3)$$

Функція (2.4) вертикального прямокутника:

$$H_{\text{vertical}}(x, y) = H(x + w, y) - H(x, y), \quad (2.4)$$

Де:

- $H(x, y)$  - Нагг-оглядова функція, що вимірює різницю сум пікселів у двох прямокутних областях.
- $A$  та  $B$  - це прямокутні області на зображенні з різними розмірами та положеннями.
- $x$  та  $y$  - координати верхнього лівого кута прямокутної області.
- $w$  та  $h$  - ширина та висота прямокутної області відповідно.

Histogram of Oriented Gradients (HOG): HOG використовує аналіз градієнтів зображення для створення дескрипторів об'єктів, що дозволяє ефективно визначати межі об'єктів.

HOG ділить зображення на малі частини, розраховує градієнти в кожній частині та створює гістограми орієнтацій градієнтів.

Дескриптори об'єктів відображають орієнтації та градієнти зображення. Це дозволяє ефективно визначати контури об'єктів.

Крім того завдяки пристосованості до освітлення його можна використовувати в умовах зміни освітлення.

Наприклад, у системах реального часу HOG може застосовуватися для виявлення пішоходів на дорозі у водійських допоміжних системах або для виявлення обличчя у системах безпеки.

Завдяки технологіям, які узагальнюють інформацію про орієнтації та градієнти зображення дескриптори допомагають ефективно описати особливості об'єктів, такі як контури. Це дозволяє визначати границі та структуру об'єктів, роблячи їх менш чутливими до змін у розмірі та розташуванні.

HOG базується на ідеї, що контури та форми об'єктів можна визначити за їх градієнтами і орієнтаціями в зображенні. Для кожного пікселя обчислюється градієнт, а потім зображення розбивається на малі блоки, для яких обчислюється гістограма орієнтацій градієнтів. Ці гістограми, в сутності, створюють дескриптори, що представляють структуру об'єкта. Цей метод дозволяє стабільно виявляти контури об'єктів навіть при зміні масштабу чи позиції.

Однією з переваг HOG є його здатність працювати в умовах змінного освітлення. Оскільки він базується на аналізі градієнтів, а не на точному вимірюванні яскравості пікселів, HOG дещо менш чутливий до варіацій у рівнях освітлення. Це дозволяє застосовувати його у різних умовах освітлення без втрати точності об'єктного виявлення.

Одним з ключових математичних аспектів методу HOG є використання градієнтів і їхніх орієнтацій для створення гістограм, які узагальнюють важливу інформацію зображення. Це дозволяє зберегти структурні особливості об'єктів, що забезпечує стійкість методу до різних умов освітлення та масштабу.

І для цих цілей використовуються наступні формули:

Горизонтальний градієнт:

$$G_x = I(x + 1, y) - I(x - 1, y), \quad (2.5)$$

Визначає різницю яскравості між сусідніми пікселями на горизонтальному напрямку. Це дозволяє виявити зміни яскравості вздовж горизонталі.

Вертикальний градієнт:

$$G_y = I(x, y + 1) - I(x, y - 1), \quad (2.6)$$

Подібно горизонтальному градієнту, визначає різницю яскравості між вертикальними сусідніми пікселями. Це спрямовано на виявлення змін яскравості вздовж вертикалі.

Магнітуда градієнту:

$$\text{Magnitude} = \sqrt{G_x^2 + G_y^2}, \quad (2.7)$$

Об'єднує горизонтальні та вертикальні градієнти для визначення магнітуди загального градієнту в кожній точці зображення.

Орієнтація градієнту:



$$\text{Orientation} = \arctan \frac{G_y}{G_x}, \quad (2.8)$$

Обчислює кутове значення орієнтації градієнту, що вказує напрямок зміни яскравості в точці зображення.

### **2.1.2 Методи детектування об'єктів з використанням глибокого навчання**

Методи детектування об'єктів з використанням глибокого навчання рис. 2.2 представляють широкий спектр підходів, які базуються на використанні нейронних мереж для виявлення об'єктів на зображеннях або відео.

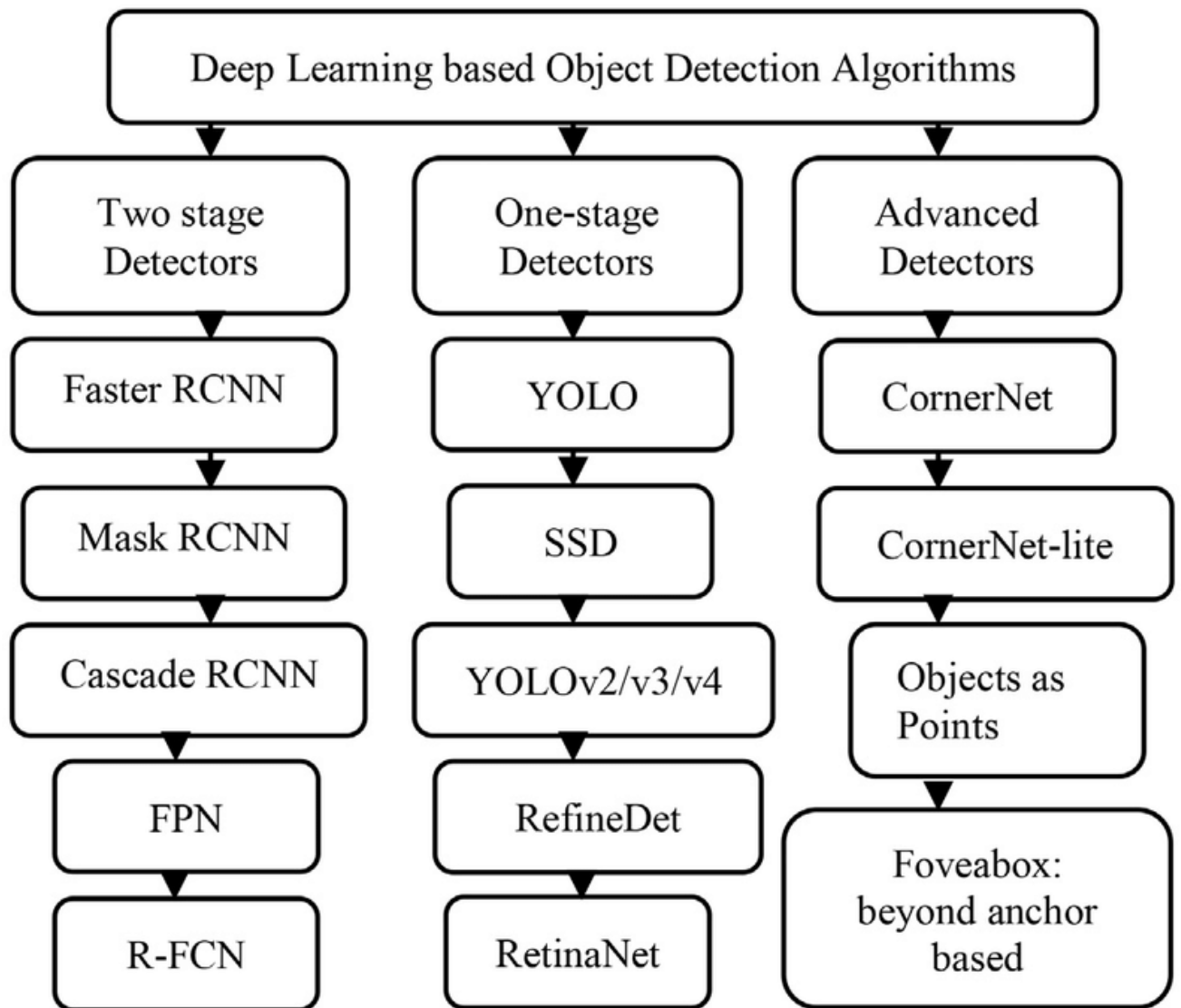


Рис. 2.2 Базові алгоритми глибокого навчання для відслідковування об'єктів [18]

Згорткові нейронні мережі (CNN) є одними з найпоширеніших моделей для детектування об'єктів. Вони мають здатність автоматично відбирати та навчатися властивостям об'єктів на зображеннях, використовуючи конволюційні шари для виявлення візуальних ознак.

Single Shot Multibox Detector (SSD) — це архітектура, яка комбінує у собі ефективність та точність виявлення об'єктів. Вона використовує згорткову мережу для прогнозування різних класів та координат об'єктів безпосередньо на різних шарах мережі.

Region-based Convolutional Neural Networks (R-CNN) використовує регіони пропозицій об'єктів та виконує класифікацію та визначення меж кожного об'єкта в окремих областях зображення. Мережа визначає регіони, які мають потенційно містити об'єкти, а потім аналізує їх детальніше для визначення класів та положення об'єктів.

Ці методи забезпечують високу точність виявлення об'єктів і є основою для багатьох сучасних систем комп'ютерного зору. Проте вони вимагають значних обчислювальних ресурсів та даних для навчання та ефективності.

Ось кілька основних речей необхідних CNN для детектування об'єктів:

- Навчання моделі. Створення навчального набору даних: потрібно зібрати достатньо зображень об'єктів, які потрібно детектувати, та розмітити їх (позначити місця на зображеннях, де знаходяться ці об'єкти).
- Тестування та застосування. Виявлення об'єктів на нових зображеннях: Після навчання модель може бути використана для виявлення об'єктів на нових зображеннях. Мережа аналізує зображення, використовуючи навчені шаблони та ознаки для виявлення ймовірності наявності об'єктів.
- Архітектура мережі. Деякі архітектури CNN спеціально розроблені для завдань детектування об'єктів, наприклад, Region-based CNN (R-CNN), Fast R-CNN, Faster R-CNN, або їх удосконалені версії. Вони використовують пропозиції областей (Region Proposals) та інші техніки для виявлення об'єктів у реальному часі.
- Застосування у відеоаналітиці. Крім зображень, CNN можуть бути використані для виявлення об'єктів у відео. Алгоритми, побудовані на основі CNN, можуть працювати в режимі реального часу, відстежуючи об'єкти на відеопотоці.

Основні характеристики:

CNN базуються на концепції конволюційних шарів, що дозволяє їм виявляти локальні особливості на зображеннях. Вони використовують ядра фільтрів для просування по зображенню, виокремлюючи властивості на різних рівнях абстракції.

Математично, це представляється формулою:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) \cdot K(i - m, j - n), \quad (2.9)$$

Де:

$S(i, j)$  - значення пікселя у вихідному зображенні,

$I$  - вхідне зображення,

$K$  - ядро (фільтр).

Пулінгові шари зменшують просторові розміри отриманої карти ознак шляхом об'єднання (пулінгу) значень з певних областей. Це дозволяє зменшити кількість параметрів та оброблення, при цьому зберігаючи важливі властивості.

Математично це може бути представлено, наприклад, через максимальне об'єднання (max-pooling):

$$M(i, j) = \max(I(i, j)), \quad (2.10)$$

Де:

$M$  - вихідне зображення після підсумкового шару,

$I$  - вхідне зображення.

Повні з'єднані шари використовуються для кінцевої класифікації об'єктів на зображенні. Вони отримують на вході вектори ознак з попередніх шарів та генерують прогнози про класи об'єктів.

Математично це представляється у вигляді лінійного перетворення:

$$Wx + b, \quad (2.11)$$

де

$W$  - матриця вагів, яка представляє внутрішні параметри моделі;

$x$  - вектор вхідних даних;

$b$  - вектор зсуву (bias), який додається після множення на матрицю вагів.

### 2.1.3 Детектування об'єктів у багатокамерних системах

#### 1. Мультифреймові методи рис. 2.3.

Цей підхід використовує дані, отримані з різних камер, для аналізу об'єктів в просторі. Здійснюючи аналіз на основі даних з різних ракурсів, система може поліпшити точність виявлення об'єктів та їхню трекінгову стабільність. Припустимо, використання трьох камер для виявлення об'єкта може дати додаткову інформацію про його розташування в просторі, забезпечуючи системі більш точні дані для визначення шляху руху або просторових взаємозв'язків.

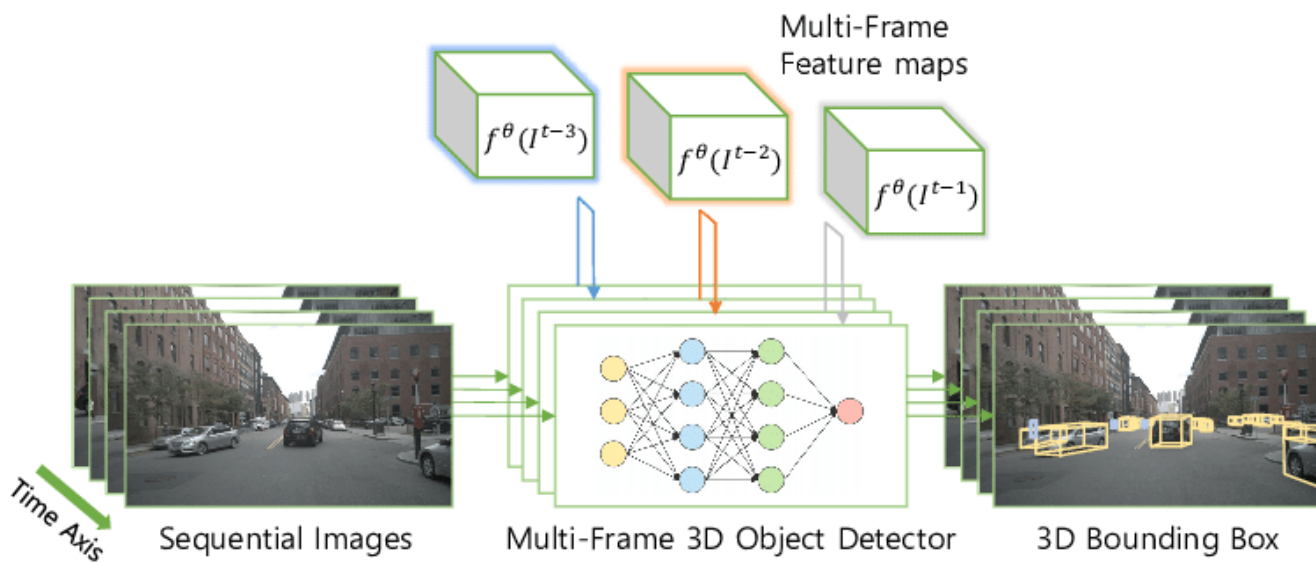


Рис. 2.3 Приклад мультифреймового методу [23]

## 2. Методи об'єднання результатів рис. 2.4.

Ці методи спрямовані на поєднання інформації з різних джерел або камер для покращення точності виявлення об'єктів. Припустимо, якщо одна камера забезпечує зображення в низькому освітленні, а інша - високою чіткістю, техніки об'єднання дозволяють об'єднати ці дані для отримання більш точної та повної інформації про об'єкт, навіть у складних умовах.

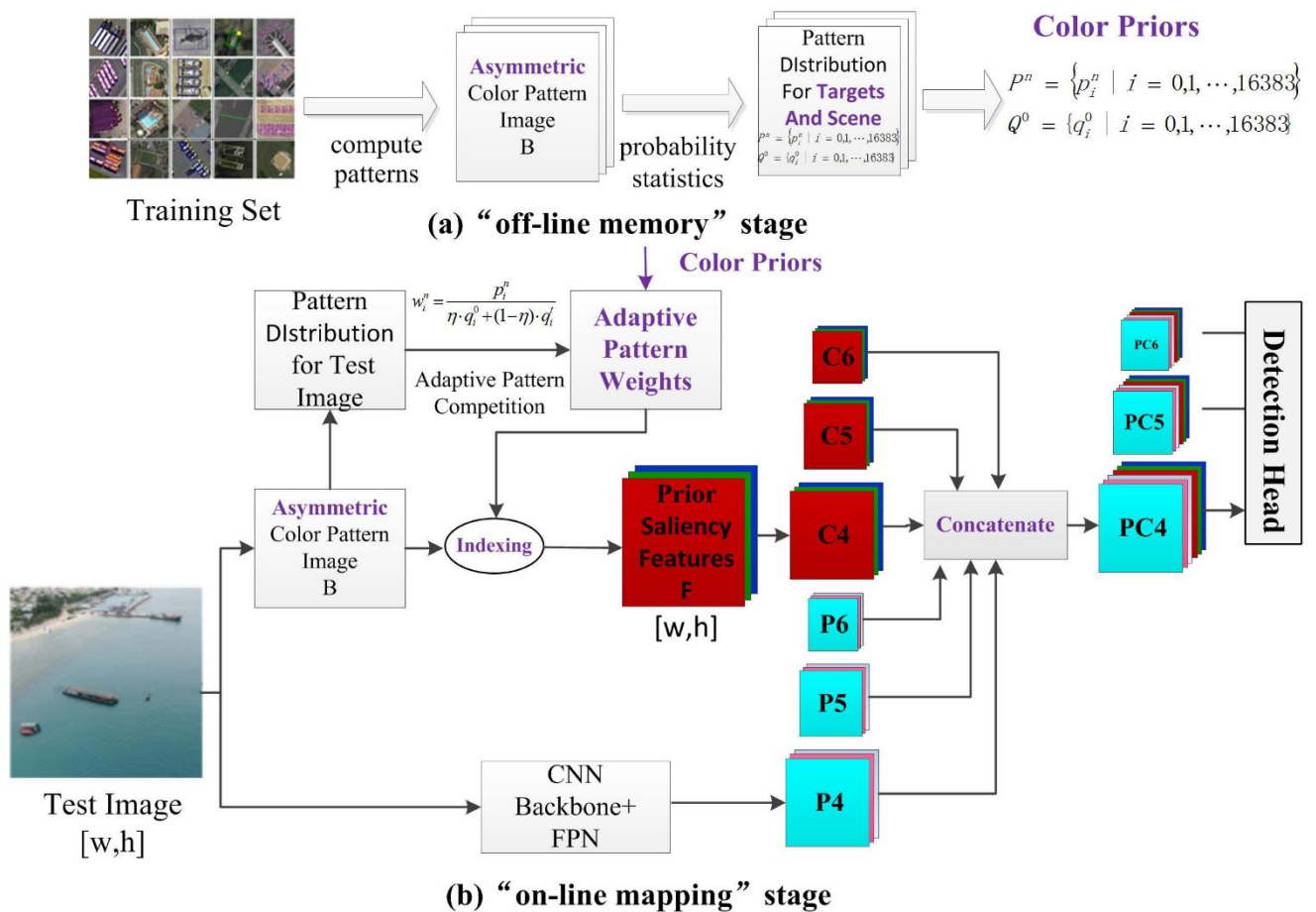


Рис. 2.4 Приклад методу об'єднання результатів [27]

## 3. Глибоке навчання для мультикамерних систем рис. 2.5.

Це використання глибоких нейронних мереж для обробки даних, зібраних з багатьох камер. Використання глибокого навчання дозволяє автоматизувати процес обробки цих об'ємів інформації, виявлення об'єктів та їхнього трекінгу. За

належного тренування моделей на відповідних наборах даних це може забезпечити більш точні та швидкі результати.

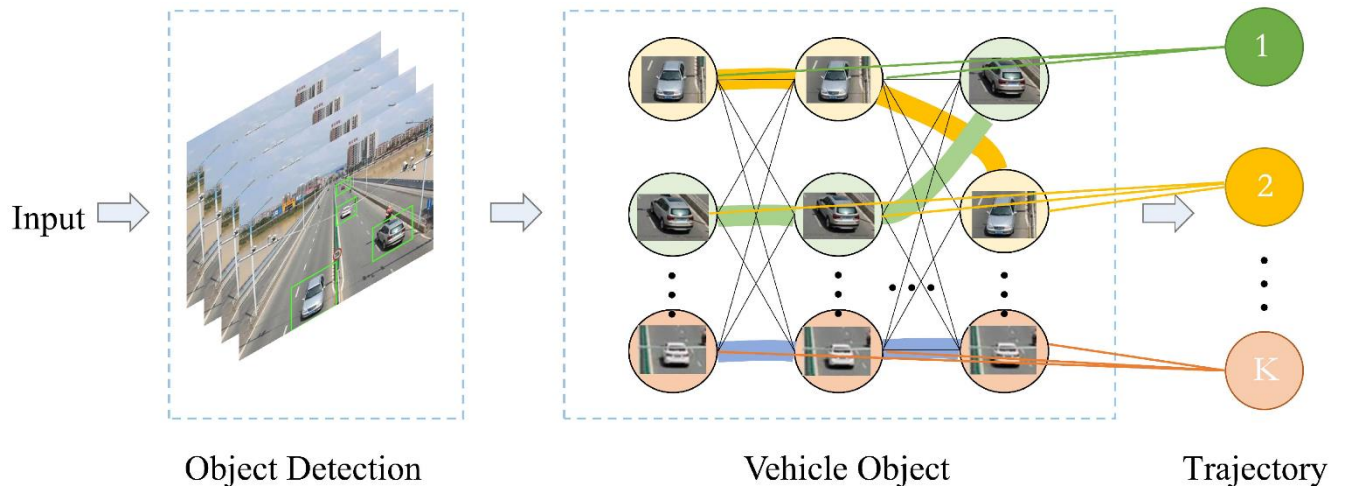


Рис. 2.5 Приклад глибокого навчання для мультикамерних систем [23]

## 2.2 Аналіз переваг та недоліків методів детектування об'єктів при багатоканальному трекінгу на основі глибокого навчання .

Глибоке навчання дозволяє створювати моделі, які мають надзвичайну точність у виявленні об'єктів навіть у складних умовах, наприклад, в умовах поганого освітлення або при перекритті об'єктів. Наприклад, системи автономних автомобілів, що використовують CNN для виявлення пішоходів або інших транспортних засобів, виявляються надзвичайно точними в реальному часі.

Методи глибокого навчання автоматично визначають корисні ознаки у вхідних даних, без необхідності ручної настройки чи вибору ознак. Це дозволяє моделям самостійно вивчати корисні ознаки у зображеннях, що покращує їх універсальність та робить їх придатними для різних завдань, як, наприклад, виявлення облич чи транспортних засобів у відеопотоці.

Глибокі нейронні мережі можуть використовуватися для різноманітних завдань, наприклад, виявлення об'єктів на зображеннях, відеопотоках чи навіть у техніці обробки медичних зображень для виявлення патологій.

Глибоке навчання ефективно використовує великі набори даних для навчання моделей. Це дозволяє отримувати більш точні та загальні моделі. Наприклад, для виявлення об'єктів у великих наборах відеоданих.

Існує постійний прогрес у глибокому навчанні, нові архітектури та методи навчання постійно розвиваються, що сприяє покращенню точності та швидкості детектування об'єктів.

Ці переваги дозволяють глибоким нейронним мережам стати потужним інструментом для виявлення об'єктів у багатокамерних системах, забезпечуючи високу точність, універсальність та здатність до роботи з великими обсягами даних.

Глибоке навчання потребує великих обсягів даних для ефективного навчання. У випадках обмеженого доступу до великих обсягів даних або коли дані мають певні обмеження (наприклад, в техніці медичного зображення), може бути важко створити достатньо репрезентативний набір даних для навчання надійних моделей.

Глибокі моделі, навчені для конкретних об'єктів або умов, можуть бути менш ефективними при розпізнаванні нових об'єктів чи у нових умовах, яких не було у навчальних даних.

Деякі глибокі моделі можуть бути дуже складними та вимогливими до обчислювальних ресурсів, особливо при роботі в реальному часі. Це може обмежувати їх застосування у вимогливих до швидкості системах, наприклад, у реальному часі для систем безпеки або транспорту.

Навчальні дані повинні бути правильно розмічені для навчання моделі. Ручне розмічення може бути часомістким та підвищувати вартість процесу підготовки даних.

Використання глибокого навчання може породжувати питання щодо етики та захисту приватності даних. Моделі, навчені на основі великих обсягів особистих даних, можуть викликати сумніви щодо конфіденційності та етики.



Ці недоліки ставлять певні виклики перед застосуванням глибокого навчання у багатокамерних системах та вимагають уважного урахування при проектуванні та впровадженні таких систем.

Для порівняльної характеристики зобразимо таблицю 2.1:

Таблиця 2.1

Порівняння методів детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання

Характеристика	HOG	CNN	Viola-Jones Algorithm	YOLO (You Only Look Once)	SSD (Single Shot Detector)	R-CNN (Region-based Convolutional Neural Networks)
Ефективність	Середня	Висока	Низька	Висока	Висока	Висока
Обсяг даних	Невеликий	Великий	Середній	Великий	Великий	Великий
Обчислювальна складність	Низька	Висока	Середня	Висока	Висока	Висока
Точність детектування	Добре для простих об'єктів	Висока точність для складних об'єктів	Низька	Дуже висока	Висока	Висока
Розмічення даних	Вимагає менше часу	Вимагає багато часу	Потребує менше часу	Вимагає багато часу	Вимагає багато часу	Вимагає багато часу
Швидкість роботи	Висока	Висока	Низька	Висока	Висока	Середня

Статистичні округлені дані для цих методів представлені на таблиці 2.2.

Таблиця 2.2

## Порівняння методів глибокого навчання

Метод	Точність (%)	Швидкість (%)	Обчислювальна складність (%)	Надійність (%)	Потужність (%)
Viola-Jones	85	70	80	90	75
Histogram of Oriented Gradients	88	60	85	80	70
YOLO (You Only Look Once)	92	50	90	85	80
Faster R-CNN	95	40	95	90	85
SSD (Single Shot MultiBox Detector)	93	45	92	85	82

### **2.3 Математична модель методу детектування об'єктів при багатоканальному трекінгу на основі глибокого навчання.**

Математична модель в науковій роботі - це абстрактне представлення реальної системи, процесу чи явища за допомогою математичних виразів, рівнянь, графіків, алгоритмів та символів. Вона використовується для опису, розуміння, прогнозування та аналізу різних явищ у природі, техніці, економіці та інших галузях.

Основні характеристики математичної моделі:

1. Узагальнення реальності. Математична модель спрощує складну реальну систему, але залишає основні аспекти для подальшого дослідження.
2. Формалізація взаємозв'язків. Вона виражає взаємозв'язки між різними компонентами системи чи процесу за допомогою математичних співвідношень.
3. Прогнозування та аналіз. Математичні моделі дозволяють робити прогнози, тестувати різні сценарії, аналізувати вплив різних факторів та варіантів на систему.
4. Ефективність. Вони дозволяють здійснювати різні обчислення та висновки шляхом використання математичних операцій та алгоритмів.

У науковій роботі, зокрема у контексті дослідження методу детектування об'єктів при багатоканальному трекінгу на основі глибокого навчання, математична модель може описувати архітектуру нейронної мережі, формально визначати параметри алгоритмів обробки зображень, представляти способи обробки даних з різних камер, а також визначати критерії оцінки результатів та точності детектування об'єктів.

Важливо враховувати, що математична модель є лише спрощеним відображенням реальності, і вона може мати обмеження та припущення, які

потрібно ураховувати при її застосуванні та інтерпретації результатів дослідження.

У своїй роботі я працював над системою з багатьох камер для формування тривимірного простору і відповідно використовував математичні моделі пов'язані з цією особливістю.

Розглянемо приклад роботи з системою відслідковування об'єктів у контексті відслідковування руху м'яча у спортивних відео.

Математичні моделі та формул, які можуть застосовуватися в обраній системі спостережень:

#### 1. Модель відстеження об'єктів в двовимірному просторі:

- Дискретна модель руху. Можна використовувати формули для обчислення прогнозованого руху об'єкта на основі попередніх позицій та швидкостей.

$$p_t = p_{t-1} + v_{t-1} \cdot \Delta t + \frac{1}{2} a \cdot \Delta t^2, \quad (2.12)$$

де:

$p_t$  - нова позиція в момент часу

$p_{t-1}$  - попередня позиція в момент часу

$v_{t-1}$  - швидкість в момент часу  $t-1$

$a$  - прискорення (вважається сталим)

$\Delta t$  - крок часу між  $t$  та  $t-1$

Ця формула передбачає рух без урахування впливу зовнішніх сил, вітру або суперечливих факторів. Для складніших моделей руху можуть використовуватись

більш складні диференціальні рівняння або методи апроксимації руху об'єктів у тривимірному просторі.

- Формули фільтра Калмана для корекції та прогнозування позицій об'єкта з урахуванням помилок вимірювань та шумів.

Формула (2.13) - прогноз позиції (прогнозована оцінка):

$$\widehat{x}_k = F \cdot \widehat{x}_{k-1} + B \cdot u_k, \quad (2.13)$$

Формула (2.14) - прогноз похибки прогнозу:

$$P_k = F \cdot P_{k-1} \cdot F^T + Q, \quad (2.14)$$

Формула (2.15) - оцінка помилки вимірювання:

$$K_k = P_k \cdot H^T \cdot (H \cdot P_k \cdot H^T + R)^{-1}, \quad (2.15)$$

Формула (2.16) - корекція позиції на основі вимірювань:

$$\widehat{x}_k = \widehat{x}_k + K_k \cdot (z_k - H \cdot \widehat{x}_k), \quad (2.16)$$

Формула (2.17) - корекція похибки прогнозу:

$$P_k = (I - K_k \cdot H) \cdot P_k, \quad (2.17)$$

$\hat{x}_k$  - оцінка стану системи в момент часу  $k$  після корекції (прогнозована позиція)

$P_k$  - матриця коваріації помилок оцінок в момент часу  $k$  після корекції (оцінка похибки прогнозу)

$F$  - матриця перехідних коефіцієнтів

$B$  - матриця коефіцієнтів управління

$u_k$  - вектор вхідних сигналів у момент часу  $k$

$Q$  - матриця коваріації шуму процесу

$K_k$  - матриця Калмана в момент часу  $k$  (Коефіцієнт корекції)

$H$  - матриця вимірювання

$R$  - матриця коваріації шуму вимірювання

$z_k$  - вектор вимірів в момент часу  $k$

2. Модель відстеження об'єктів у тривимірному просторі:

- Стереозорієнтовані моделі. Формули для реконструкції тривимірного простору на основі зображень з різних камер та обчислення глибини.

Триангуляція за допомогою стереопар камер:

$$Z = \frac{f \cdot T}{d}, \quad (2.18)$$

Де:

$Z$  - глибина об'єкта;

$f$  - фокусна відстань камери;

$T$  - база (відстань між камерами);

$d$  - зсув (диспаратет) між відповідними точками на двох зображеннях.

Ця формула ґрунтується на принципі триангуляції, де за допомогою інформації з двох зображень та їхніх параметрів можна обчислити глибину об'єктів у тривимірному просторі.

Алгоритми реконструкції на основі співставлення особливостей (feature matching):

SIFT (Scale-Invariant Feature Transform):

Детектор. Диференціальне представлення гаусіанських пірамід:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad (2.19)$$

Ця формула описує конволюцію  $G$  (гаусіанської функції) з зображенням  $I$  для створення гаусіанської піраміди.

Спрощення детектора локальних екстремумів:

$$D(x) = L(x) - \max(L(x - 1), L(x), L(x + 1)), \quad (2.20)$$

Ця формула означає віднімання значення центрального пікселя  $L(x)$  від максимального значення в околі, для виявлення локальних екстремумів після створення гаусіанської піраміди.

SURF (Speeded-Up Robust Features):

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}, \quad (2.21)$$

Ця матриця містить другі похідні гаусіанської функції для обчислення особливостей. А також обчислення відповідності для точок особливостей.

ORB (Oriented FAST and Rotated BRIEF):

FAST (Features from Accelerated Segment Test)

$$\text{Corner response} = |I(x) - I(x_c)| - \text{threshold}, \quad (2.22)$$

Де:

$I(x)$  представляє інтенсивність пікселя у точці  $x$ ;

$I(x_c)$  представляє інтенсивність центрального пікселя;

threshold - порогове значення, яке визначає, коли піксель вважається ключовою точкою.

$$B(i, j) = \begin{cases} 1, & \text{if } I(p_i) < I(p_j) \\ 0, & \text{otherwise} \end{cases}, \quad (2.23)$$

Де:

$B(i, j)$  - бінарний біт для порівняння інтенсивностей двох пікселів;

$I(p_i)$  та  $I(p_j)$  - інтенсивності пікселів  $p_i$  та  $p_j$ .

Вони базуються на виявленні та порівнянні особливостей (точок із унікальними характеристиками) на різних зображеннях, після чого обчислюється їхня просторова взаємодія для реконструкції тривимірного простору.



Використання глибинних карт:

$$Z(x, y) = \frac{f \cdot T}{d(x, y)}, \quad (2.24)$$

Де:

$Z(x, y)$  - глибина пікселя на координатах  $(x, y)$ ;

$f$  - фокусна відстань камери;

$T$  - база (відстань між камерами);

$d(x, y)$  - диспаратет для конкретного пікселя.

- Геометричні вирази. Можуть використовуватися геометричні формули для обчислення та оновлення позицій об'єктів у тривимірному просторі на основі даних з різних камер.

Геометрична формула для обчислення глибини об'єкта:

$$Z = \frac{f \cdot T}{d}, \quad (2.25)$$

У цій формулі:

$Z$  - глибина об'єкта;

$f$  - фокусна відстань камери;

$T$  - відстань між камерами (база);

$d$  - диспаратет між відповідними точками на двох зображеннях.

Ця формула використовує різницю між зображеннями з різних камер для визначення глибини об'єктів, базуючись на принципах геометрії трикутників.

Геометрична формула для визначення просторової позиції об'єкта на основі даних з камер:

$$X = P \cdot x, \quad (2.26)$$

У цій формулі:

$X$  - тривимірний вектор позиції об'єкта;

$P$  - матриця проєкції камери;

$x$  - двовимірний вектор позиції на площині зображення. Ця формула використовує матрицю проєкції камери для перетворення двовимірної позиції на площині зображення в тривимірну просторову позицію об'єкта.

3. Алгоритми для покращення виявлення та відстеження:

- Багатомасштабні функції. Формули для обробки зображень у різних масштабах для поліпшення точності виявлення м'яча.

Гаусівська піраміда використовує гаусівську фільтрацію для створення послідовності зображень різної роздільної здатності. Формула для створення рівнів піраміди може бути подана як:

$$I_{k+1} = \frac{1}{4} (I_k * G(\sigma)) * G(\sigma), \quad (2.27)$$

Тут:

$I_{k+1}$  - це новий рівень гаусівської піраміди, який ми створюємо.

$I_k$  - це попередній рівень гаусівської піраміди.

\* - означає операцію згладжування, яку ми виконуємо з використанням фільтру Гауса.

$G(\sigma)$  - це фільтр Гауса з параметром  $\sigma$ , який визначає ступінь розмиття.

Гаусівська піраміда рис. 2.6 дозволяє аналізувати зображення на різних рівнях деталізації, що може поліпшити виявлення об'єктів за рахунок зменшення шуму та підвищення точності.

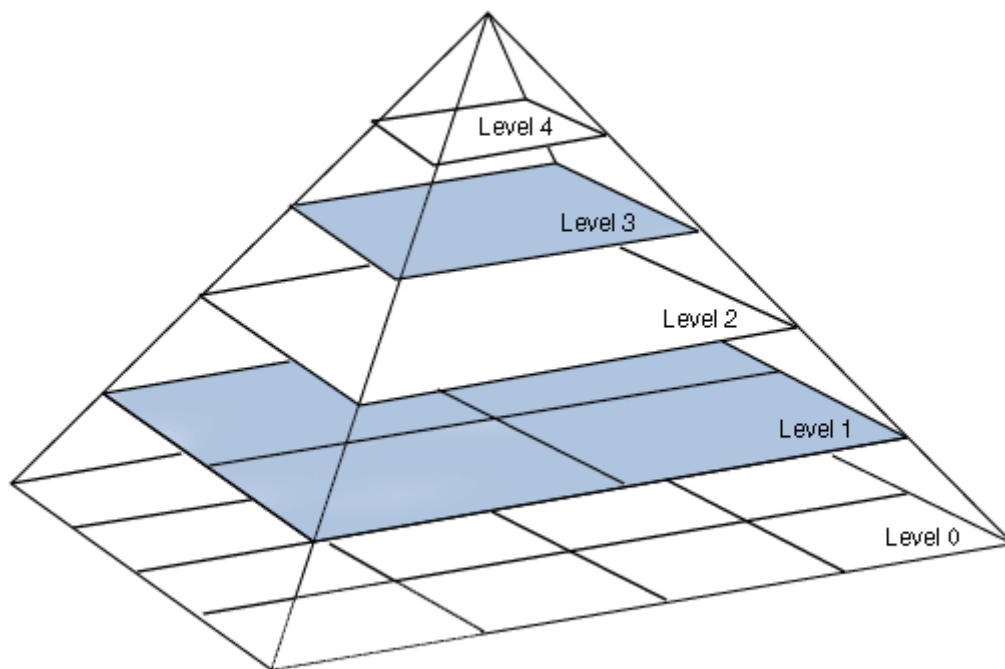


Рис. 2.6 Приклад Гаусівської піраміди для зображень [8]

Або

$$I_{k+1} = \frac{1}{2} (I_k * G(\sigma)) \oplus L_{k+1}, \quad (2.28)$$

Де:

$I_{k+1}$  - це новий рівень гаусівської піраміди, який ми створюємо.

$\frac{1}{2}(I_k * G(\sigma))$  - це попередній рівень гаусівської піраміди ( $I_k$ ), який ми згладжуємо за допомогою фільтра Гауса і після цього зменшуємо розмірність вдвічі (відтак коефіцієнт  $\frac{1}{2}$ )

$\oplus$  - це операція підвищення розмірності, яка може використовуватися для врахування додаткового компонента  $I_{k+1}$ , що представляє додаткову інформацію або деталі при підвищенні розмірності.

Цей вираз застосовує фільтр Гауса  $G$  до початкового зображення  $I$  для створення піраміди зображень на різних масштабах. Це дозволяє виявляти об'єкти за різних розмірів, поліпшуючи точність виявлення, оскільки деякі об'єкти можуть бути видимими тільки на певних масштабах зображення.

Геометричний вираз для багатомасштабного виявлення м'яча на зображенні:

$$R(x, y, s) = \sum_{i=-w}^w \sum_{j=-w}^w G(i, j, s) \cdot |I(x + i, y + j) - I(x, y)|^2, \quad (2.29)$$

Де:

$R(x, y, s)$  - значення багатомасштабної функції в точці  $(x, y)$  і масштабі  $s$ ;

$G(i, j, s)$  - гаусіанська вагова функція з параметрами  $(i, j, s)$ ;

$I(x, y)$  - інтенсивність пікселя на зображенні.

Цей вираз описує ваговану суму квадратів різниць інтенсивностей між пікселями на різних масштабах, що допомагає виявити м'яч у різних розмірах. Застосування гаусіанської вагової функції дозволяє зменшити вплив шумів і виокремити структури на різних масштабах. Сумування в квадраті різниць інтенсивностей дозволяє визначити області з великою зміною яскравості, що може свідчити про наявність м'яча.

- Перехресний огляд для обробки оклюзій. Математичні методи для аналізу даних з різних джерел для вирішення проблеми оклюзій.

Математичний вираз для перехресного огляду для обробки оклюзій:

$$L(x, y, s) = \max_{i,j} R(x + i, y + j, s), \quad (2.30)$$

Де:

$L(x,y,s)$  - значення перехресного огляду в точці  $(x,y)$  і масштабі

$R(x+i,y+j,s)$  - значення багатомасштабної функції в точці  $(x+i,y+j)$  і масштабі  $s$ .

Цей вираз визначає значення перехресного огляду як максимальне значення багатомасштабної функції в області навколо даної точки. Це допомагає вирішити проблему оклюзій, коли об'єкт частково перекривається іншим об'єктом на одному зображенні. Перехресний огляд використовує інформацію з різних джерел (різних масштабів) для визначення найважливіших ознак в області точки. Використання максимуму дозволяє врахувати найсильніші ознаки з різних джерел, що допомагає вирішити проблему оклюзій і поліпшити точність виявлення об'єкта.

#### 4. Оптимізація траєкторій та об'єднання позицій у тривимірному просторі:

- Методи оптимізації. Формули для покращення точності позицій об'єктів у тривимірному просторі, наприклад, методи мінімізації помилок.

Математичний вираз для мінімізації помилок у тривимірному просторі може бути представлений формулою фільтрації, наприклад, фільтром Калмана:

$$\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k(z_k - H_k \widehat{x}_{k|k-1}), \quad (2.31)$$

У цій формулі:

$\widehat{x}_{k|k}$  - корегована оцінка поточного стану;

$\widehat{x}_{k|k-1}$  - прогнозована оцінка попереднього стану;

$K_k$  - коефіцієнт Калмана для коригування прогнозу;

$z_k$  - вимірювання в момент часу  $k$ ;

$H_k$  - матриця спостереження.

Фільтр Калмана є рекурсивним алгоритмом оптимізації, який поєднує в собі інформацію з вимірювань та попередніх прогнозів для покращення точності оцінки стану системи. Ця формула використовується для коригування попередньої оцінки стану об'єкта з урахуванням нових вимірювань, що дозволяє поліпшити точність прогнозування позиції об'єкта у тривимірному просторі.

- Фільтр Калмана для тривимірного відстеження. Аналогічні методи фільтра Калмана, але вже у тривимірному просторі для покращення точності відстеження.

Прогнозований стан:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k, \quad (2.32)$$

Де:

$\hat{x}_{k|k-1}$  - прогнозований стан в момент часу

$F_k$  - матриця перехідного процесу;

$\hat{x}_{k-1|k-1}$  - корегований стан на попередньому кроці;

$B_k$  - матриця управління;

$u_k$  - вхідні керуючі сигнали.

Оцінка помилок прогнозування коваріації:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k, \quad (2.33)$$

Де:

$P_{k|k-1}$  - оцінка помилок прогнозування коваріації в момент часу  $k$

$Q_k$  - матриця коваріації випадкового шуму процесу.

Координата об'єкта:

$$z_k = H_k \hat{x}_{k|k-1} + v_k, \quad (2.34)$$

Де:

$z_k$  - вимірювання координати об'єкта в момент часу  $k$

$H_k$  - матриця спостереження;

$v_k$  - випадковий шум вимірювання.

Оцінка помилок вимірювання коваріації:

$$R_k = E(v_k v_k^T), \quad (2.35)$$

Де:

$R_k$  - матриця коваріації помилок вимірювання.

Ваговий коефіцієнт Калмана:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}, \quad (2.36)$$

Де:

$K_k$ - ваговий коефіцієнт Калмана.

Корегований стан:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1}), \quad (2.37)$$

Ця формула використовує ваговий коефіцієнт Калмана для коригування прогнозованого стану з урахуванням нового вимірювання.

Оцінка помилок корекції коваріації:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}, \quad (2.38)$$

$P_{k|k}$  - оцінка помилок корекції коваріації.

Фільтр Калмана рис. 2.7 для тривимірного відстеження використовує матриці та коефіцієнти для прогнозування та корекції стану системи. Оновлення стану враховує інформацію з вимірювань та ваговий коефіцієнт Калмана для підтримки балансу між прогнозуванням та вимірюваннями для поліпшення точності відстеження у тривимірному просторі.



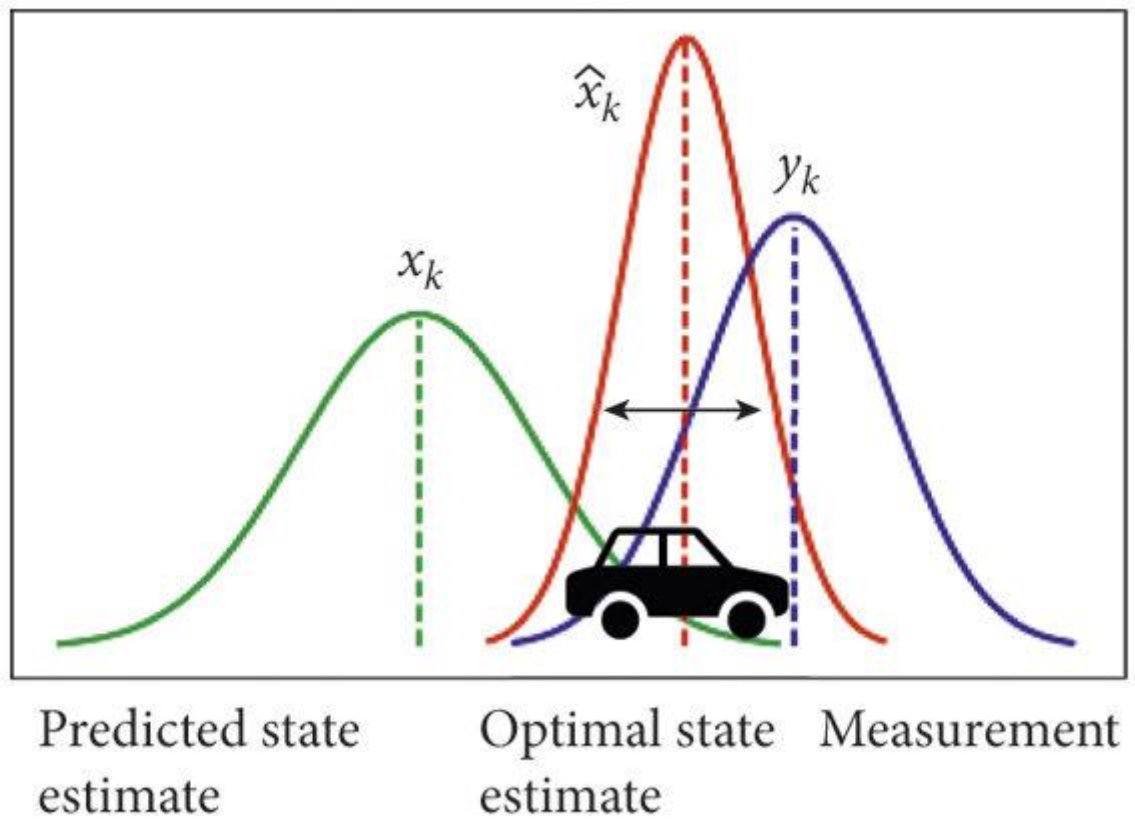


Рис. 2.7 Зображення принципів роботи фільтра Калмана [7]

## **3 РОЗРОБКА МЕТОДУ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ ПРИ БАГАТОКАМЕРНОМУ ТРЕКІНГУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ**

### **3.1 Опис розробки методу**

Задача полягає у локалізації тривимірного положення м'яча в умовах синхронізованих послідовностей зображень, отриманих з різних камер. Необхідно розв'язати завдання визначення положення м'яча у тривимірному просторі на основі вхідних даних, що включають послідовності зображень для кожної камери в певний момент часу. Метою є розробка ефективного методу мультикамерного трекінгу об'єктів з використанням глибокого навчання для підвищення точності та стійкості виявлення м'яча в умовах затемнення, перешкод та помилкових виявлень.

Головна мотивація запропонованої структури мультикамерного трекінгу об'єктів полягає у використанні інформативних ознак CNN для виявлення м'яча для кожної камери, а потім вивченні перехресної інформації для покращення стійкості трекінгу, особливо проти пропущених виявлень через затемнення та помилкових виявлень, спричинених шумовими перешкодами. Використовуємо типову структуру трекінгу з виявленням, прийняли Faster RCNN [15] та ECO tracker [16] як основу для виявлення та трекінгу 2D м'яча відповідно. Потім техніка триангуляції разом з матрицями калібрування камер використовується для об'єднання кількох 2D позицій м'яча в 3D місце розташування. Наостанок, застосовується фільтр Калмана для плавності траєкторії 3D м'яча.

Для зображень, отриманих від кількох камер до моменту часу  $t$ , запропонована система визначає положення м'яча у тривимірному просторі за чотирма послідовними кроками:

- Виявлення м'яча у двовимірному просторі: система виявляє м'яч у кожній камері на основі детектора Faster RCNN. Оскільки м'яч має невеликий розмір, ми використовуємо метод виявлення невеликих об'єктів, запропонований у [27], [28], та відповідно до нього вносимо правки у мережу на основі VGG16, об'єднуючи картинки ознак, отриманих з мультишарових рівнів. Хоча продуктивність виявлення м'яча задовільна, все ще виникають помилкові та пропущені виявлення через завади шуму та перекриття.
- Трекінг м'яча у двовимірному просторі: В якості базового використовується трекер ESO [6] для відстеження м'яча у кожному кадрі. У порівнянні з оригінальною версією, трекінг м'яча у двовимірному просторі має два покращення: з одного боку, деякі помилкові виявлення видаляються, а деякі пропущені компенсуються за рахунок перекриття з інших камер. З іншого боку, проблему відслідковування шаблону вирішуємо, оновлюючи модель відстеження залежно від виявлення.
- Об'єднання позицій у тривимірному просторі: Кілька позицій м'яча у двовимірному просторі об'єднуються у тривимірне положення за допомогою методу триангуляції [29]. Для пар камер  $i$  та  $j$  вихід триангуляції включає тривимірне положення та відповідну похибку. Похибку вимірюємо за допомогою відстаней репроекції між вихідними точками зображення та точками репроекції вище 3D-точки. Завершене тривимірне положення м'яча отримуємо шляхом усереднення, а відповідні похибки знаходяться в межах заданого порогу.
- Трекінг м'яча у тривимірному просторі: Фільтр Калмана [30] застосовується для поліпшення плавності трекінгу м'яча у тривимірному просторі. Запропонуємо спрощену модель руху м'яча в горизонтальній площині та вертикальному напрямку,

відповідно. Нарешті, результати тривимірного трекінгу на основі покращеного фільтра Калмана з вищезазначеним припущенням руху можна отримати за допомогою доповнення у вигляді CNN.

Загальні детектори об'єктів зазвичай класифікуються на два типи: двоступеневий та одноступеневий. Для двоступеневого детектора, це зазвичай CNN, тоді як одноступеневий детектор передбачає головним чином YOLO [2], [3] та SSD [4]. Точність та швидкість обробки є двома важливими показниками для детектора. Відомо, що детектори сімейства CNN є точнішими, тоді як YOLO та SSD працюють швидше. Оскільки для нас важлива саме точність на поточному етапі, класичний детектор CNN обраний [15] як основний.

Однак виявлено, що CNN працює добре для великих об'єктів, але не може ефективно виявляти маленькі об'єкти на зображенні. Основна причина полягає у структурі глибокої нейронної мережі, яка використовується під час вилучення ознак. Зазвичай глибока нейронна мережа складається зі згорткових та пулінгових шарів, які призводять до зменшення розміру вихідної ознаки у кілька разів порівняно з вихідним зображенням.

Насправді, для глибокої нейронної мережі ранні шари вилучають низькорівневі ознаки об'єктів, такі як лінія, край і т. д., тоді як глибші шари вилучають високорівневі ознаки з семантичного погляду. Зважаючи на те, що високорівневі семантичні ознаки мають кращі можливості розрізнення, дослідники досліджують глибші нейронні мережі для виявлення загальних об'єктів, наприклад, ResNet [6], Inception [7], GoogleNet [8], тощо. Однак ці глибокі мережі дають гірші результати для виявлення малих об'єктів через втрату інформації під час операції згортання та пулінгу, хоча вони покращують результати виявлення для великих об'єктів.

Виявлення м'яча у двовимірному просторі на основі CNN та переробленої основної мережі VGG16. Порівняно з оригінальною мережею VGG16, карти ознак, що генеруються з різних згорткових блоків, об'єднуються та спільно подаються на наступний етап.

Цей розділ розглядає архітектуру мережі для виявлення 2D м'яча, базуючись на CNN із використанням мережі VGG16 як основи. Однак, як вже було вказано, характеристики м'яча, вилучені з останнього шару оригінальної мережі VGG16, обмежені через зменшення розміру при пулінгу. Тому оригінальна мережа VGG16 не є адаптивною для виявлення малих м'ячів. Згідно з [27], мережу VGG16 удосконалено, перетворивши її на багатомасштабну мережу. Спочатку вивід з кожного згорткового блоку конкатенується та подається наступному етапу. Окрім Conv1, всі інші згорткові блоки також збільшуються до того самого розміру, що й вивід карт властивостей від Cov1. Потім всі карти ознак одного розміру конкатенуються. Операції збільшення розміру та конкатенації такі ж, як у [27]. Потім конкатеновані карти ознак надсилаються на шар областей пропозицій (RPN), і решта операцій такі ж, як у CNN. Іншими словами, області пропозицій, згенеровані з RPN, та конкатеновані карти ознак передаються на шар RoI пулінгу. Після цього карті ознак пропозицій обробляються в повністю згортковому блоку класифікації, і на виході отримуємо позицію м'яча та ймовірність його наявності.

У даній роботі спочатку зібрано набір даних і вручну позначено м'яч прямокутником. У завданні виявлення брали участь лише два класи: м'яч і не-м'яч. Навчальні вибірки підготовлені відповідно до формату VOC2007. Під час реалізації, окрім удосконалення основної мережі, деякі методи позначення можуть бути вибірковими для кращої ефективності виявлення. Вони узагальнені наступним чином:

- Позначати лише ті м'ячі, які чітко видно. Це дозволить уникнути помилкових виявлень та зробити детектор м'яча точнішим. Однак модель втратить здатність виявити м'яч, який частково чи повністю перекритий. Враховуючи, що інформація з кількох камер може бути використана для вирішення проблеми перекриття на наступному етапі, точність є основною на цьому етапі.

- Не позначати занадто густо за короткий період часу. В іншому випадку модель буде перенавчатися через занадто багато дублювання позначень.
- Попередня обробка вихідних зображень може бути необов'язковим методом, оскільки це допомагає частково усунути вплив, спричинений фоновими перешкодами та варіаціями освітлення.

Використання інформації з різних камер для корекції виявлення м'яча на 2D засноване на епіпольярних обмеженнях. Незважаючи на значне покращення точності виявлення м'яча за допомогою методів глибокого навчання, залишаються випадки помилкових позитивних та негативних виявлень. Малий розмір м'яча та часті перекриття гравцями сприяють цим непередбаченим виявленням. Обмежені характеристики вигляду маленького м'яча часто призводять до помилкової ідентифікації інших об'єктів, схожих на м'яч, таких як взуття гравців, світлові плями чи навіть голови глядачів. Крім того, у командних видах спорту м'яч часто знаходиться під перекриттям гравців, що призводить до того, що він залишається невиявленим протягом кількох кадрів.

Один зі способів вирішення цих проблем - використання інформації з кількох камер для зменшення помилкових виявлень та виявлення пропущених. Багато існуючих алгоритмів багатокамерного відстеження ґрунтуються на калібруванні камер або епіпольярних обмеженнях для вирішення цієї проблеми.

У цьому розділі основна увага приділяється використанню інформації з різних камер для виправлення виявлення м'яча на 2D за рахунок застосування епіпольярного обмеження. Це обмеження залежить від фундаментальної матриці кожної пари камер, що описує внутрішні геометричні відносини між двома переглядами камери. Ця матриця не залежить від структури сцени, а лише від внутрішніх параметрів камери та відносного положення. Фматриця показує, що якщо точка в 3D-просторі відображається в двійкових точках двох різних видів, ці точки повинні відповідати певним обмеженням, що визначені матрицею.

Фундаментальна матриця може бути просто обчислена, якщо знайдено більше ніж сім відповідних точок в двох площинах [1]. Щоб побудувати точні геометричні відносини між двома видами відповідності вручну позначаються якомога більше точок, використовуючи сторони поля як посилання. Після цього застосовується алгоритм RANSAC [2], щоб відфільтрувати викиди, і на основі утворених точок обчислюється фундаментальна матриця. Викиди - це точки, які знаходяться відносно далеко від відповідних епіполярних ліній і не використовуються для обчислення залишкової фундаментальної матриці. Отже, отримана фундаментальна матриця стійка до помилок ручного позначення.

Мета полягає в покращенні ефективності виявлення м'яча за рахунок вилучення помилкових виявлень та компенсації пропущених виявлень за допомогою дослідження інформації з різних камер.

По-перше, важливо визначити, чи є поточне виявлення в кожному з видів справжнім позитивним чи помилковим. Це можна зробити, перевіривши, чи належить викид попереднім послідовним кадрам. У разі позитивної відповіді, поточне виявлення буде позначено як помилкове та вилучено, в іншому випадку вважається справжнім та залишається. Якщо виявлення в попередніх кадрах недоступне, можна вважати справжнім, воно близьке до всіх епіполярних ліній, що відповідають справжнім виявленням з інших видів.

По-друге, якщо помилково виявлений або визначений як помилкове в попередньому кроці, пропущене виявлення може бути скомпенсоване шляхом дослідження епіполярних ліній, що відповідають справжнім виявленням в інших видів. Якщо існують дві або більше епіполярних ліній, то точка перетину ліній може бути скомпенсованим виявленням.

Хоча більшість м'ячів у кожній камері можуть бути виявлені та виправлені на основі вищезазначених двох кроків, все ж існують пропущені виявлення, коли м'яч повністю прихований гравцями з більшості видів. Тому техніка відстеження м'яча є необхідною для вирішення проблеми повного приховання. Останнім часом методи візуального відстеження для загальних об'єктів були значно розвинені [3].

Серед них, відстежувач ЕСО є одним з найсучасніших відстежувачів на основі дискримінативних кореляційних фільтрів [16]. Він може ефективно вирішувати часткове приховання без зсуву шаблону та повне приховання протягом 1 с, коли враховуються згорткові ознаки. Крім того, він стійкий до зміни меж виділення цільового об'єкта. Таким чином, для відстеження м'яча у 2D було обрано метод на основі ЕСО.

Однак, коли використовується відстежувач ЕСО, проблемою залишається зсув шаблону через малий розмір м'яча та тривале приховання. Тому під час процесу відстеження було зроблено два основні вдосконалення на основі вихідного відстежувача ЕСО.

Виявлення у поточному кадрі використовується для моніторингу зсуву шаблону відстежування. Припускаємо, що стан відстежування - 0, 1, 2, коли відстежувач зупиняється, перезапускається або продовжується, відповідно. Якщо вихід ЕСО віддалений від виявленої позиції у поточному кадрі за певну відстань протягом трьох послідовних кадрів, він визначається як зсув шаблону. У цьому випадку відстежувач ЕСО негайно перезапускається а поточне виявлення використовується як початкова позиція м'яча. Інакше поточне виявлення вважається тимчасовим шумом, і відстежувач ЕСО продовжує свою роботу. З іншого боку, якщо відстеження ЕСО порожнє, відстежувач зупиняється і чекає на наступне свіже відстеження. Щодо поточного виявлення м'яча, якщо його не виявлено протягом короткого часу можливо через часткове приховання, відстеження продовжується з незмінними параметрами відстеження. Якщо ж м'яч не може бути виявлений протягом тривалого часу (більше 20 кадрів) через тривале приховання або його можливий виїзд з поля зору, відстеження зупиняється.

У цій частині кілька 2D відстежень м'яча з різних видів об'єднуються у 3D положення за допомогою техніки триангуляції [29]. Триангуляція означає визначення точки в тривимірному просторі, виходячи з її проєкцій на два або більше зображень. Для вирішення цієї проблеми також необхідно мати попередні



знання про матриці калібрування камер, які беруть участь. Має бути перетином двох ліній видимості згенерованих з двох відповідних точок на зображеннях. Однак вони не завжди перетинаються в тривимірному просторі через шуми, такі як спотворення об'єктива або помилки виявлення точок. Таким чином, триангуляція полягає у знаходженні 3D точки, яка відповідає виміряним точкам на зображеннях.

Зазвичай, оскільки для системи відстеження м'яча з використанням кількох камер встановлюється щонайменше три або більше камер, але об'єднати кілька результатів триангуляції в один, все ще залишається проблемою. Беручи до уваги існування шумів виявлення м'яча, не всі триангульовані 3D позиції достатньо точні. Отже, необхідно оцінювати точність отриманої 3D позиції для кожної пари камер перед подальшою операцією об'єднання.

Вводиться параметр повторної проєкції для вимірювання точності кожної 3D позиції, отриманої за допомогою триангуляції. Середнє значення двох відстаней потім використовується для оцінки точності кожної триангульованої 3D позиції, чим менше, тим краще. Додатково деякі результати триангуляції видаляються шляхом встановлення порогу для помилки репроекції. Це через те, що більша помилка репроекції зазвичай означає меншу ймовірність того, що дві спостережувані 2D позиції м'яча проєктуються з однієї і тієї ж точки в тривимірному просторі. Таким чином, одне з спостережень 2D м'яча може бути шумом і його варто видалити. Є один виняток, на який варто звернути увагу, що якщо водночас в двох видах спостерігається той самий не-м'яч як м'яч, з'явиться хибне позитивне 3D положення з малою помилкою репроекції. На щастя, цей випадок в значній мірі зменшується завдяки виявленню на основі глибокого навчання, дослідженню інформації з різних видів та проєктуванню відстеження, які були вище визначені.

Узагальнюючи 3D об'єднання положення м'яча в час  $t$ , 3D положення м'яча разом з їх помилками репроекції для кожної пари камер спочатку обчислюються на основі техніки триангуляції. Після цього 3D позиції з їх помилками репроекції,

які перевищують визначений поріг, видаляються. Нарешті можна знайти усереднені 3D значення положення м'яча.

Після об'єднання 3D позицій м'яча, ми здатні здійснювати його відстеження практично в будь-який час. Але існують деякі помилкові негативні результати, особливо коли м'яч знаходиться під заслоном гравців. Окрім цього, є також помилкові позитивні результати через накопичення похибок, введених з попередніх трьох частин. У цій частині ми маємо на меті згладити траєкторію руху 3D м'яча за допомогою фільтра Калмана [30].

Фільтр Калмана широко використовується як оптимальний лінійний метод оцінки стану, тому це хороше рішення для згладжування траєкторії руху м'яча. Значення вимірів у поточний момент часу та їх передбачене значення на попередній момент дозволяють прогнозувати поточний стан, який розглядається як оптимальний результат на поточному кадрі. У випадку з 3D відстеженням м'яча, положення м'яча у поточний момент часу  $t$  може бути оптимізоване шляхом розглядання як 3D позиції м'яча у час  $t$ , так і прогнозу Калмана на попередній час. У випадку, якщо спостереження 3D м'яча недоступне, вихід просто передбачається без корекції за допомогою моделі руху м'яча.

Найважливіша умова застосування фільтра Калмана для відстеження 3D м'яча - визначення моделі руху. Наприклад, для баскетбольного м'яча складно отримати модель руху, оскільки ним зазвичай керують гравці, і він рухається нерегулярно. Загалом, рух м'яча можна розглядати як поєднання руху в горизонтальній площині та руху у вертикальному напрямі. Для спрощення, рух м'яча в горизонтальній площині вважається рухом зі сталою швидкістю, з одного боку. З іншого боку, рух у вертикальному напрямі можна розглядати як рух зі сталою швидкістю чи рух зі сталим прискоренням, залежно від фактичного стану м'яча. Для баскетбольного м'яча, зазвичай існують чотири типи рухів: в утриманні, передача, кидок та підбирання. Рух м'яча у вертикальному напрямі розглядається як рух зі сталою швидкістю, коли він знаходиться в стані утримання або передачі, в інших випадках він вважається рухом зі сталим

прискоренням, коли м'яч кидають або підбирають. Це припущення є обґрунтованим, оскільки рух м'яча між двома послідовними кадрами є дуже малим, оскільки частота кадрів зазвичай дорівнює більше 20 к/с, що також підтверджено експериментами далі. За цим припущенням рух м'яча може бути класифікований за допомогою висоти 1 м. Це означає, що вертикальний рух м'яча розглядається як рух зі сталим прискоренням, коли висота м'яча перевищує 1 м, інакше - як рух зі сталою швидкістю. Після цього фільтр Калмана використовується для відстеження 3D м'яча. У нашій роботі це поліпшене відстеження Калмана з урахуванням цього припущення про рух називається iKalman, тоді як більш просте чи грубе відстеження Калмана з повністю лінійним припущенням руху називається просто Калманом.

«APIDIS Basketball Dataset» [4]: Набір даних був отриманий за допомогою семи камер: п'ять з них розташовані на рівні підлоги навколо корта, а дві камери розташовані над кортом. Кожна камера знімає відео з частотою 25 кадрів за секунду і роздільною здатністю 800x600. Цей набір даних містить проблеми, оскільки розташування камер не дуже сприятливе для відстеження м'яча, а також є складнощі в отриманні гарного освітлення. Проте це один із небагатьох загальнодоступних наборів даних, що включають командні види спорту.

Окрім послідовності з 1500 кадрів для тестування, набір даних містить інші кадри, які охоплюють ту ж саму гру в баскетбол через інший час. Кадри з позначеним м'ячем, крім послідовності з 1500 кадрів, підготовлені для навчання детектора м'яча, а саме послідовність з 1500 кадрів використовується для тестування. Набір даних надає лише відповідну 2D «правдивість» для 1500 кадрів, але не надає 3D «правдивість». Для подальшої оцінки точності відстеження м'яча у 3D просторі генерується 3D модель на основі запропонованого методу фузії позицій 3D, із використанням наданої 2D «правди» [9].

Графік точності [4] показує відсоток кадрів, де оцінена позиція знаходиться в межах заданої відстані від дійсної правди. Позиція відноситься до центру м'яча,

а відстань - це евклідова відстань між центром м'яча, що відстежується, і вручну позначеною дійсною правдою.

Оскільки запропонована структура включає відстеження м'яча як у 2D, так і у 3D просторі, краще оцінювати як 2D, так і 3D результати відстеження. Порівняно з 2D правдою м'яча, яку можна просто позначити вручну, отримання 3D правди вручну є майже неможливим. Навіть у такому випадку, 2D правда може бути використана для генерації віртуальної 3D правди за допомогою техніки триангуляції. Поріг 2D встановлений від 1 до 50 пікселів, в той час як поріг 3D змінюється від 1 до 1000 см.

У цій частині реалізуються різні експерименти, а їх продуктивність оцінюється як у 2D, так і у 3D просторі для демонстрації ефективності запропонованої багатокамерної системи відстеження м'яча. Окрім порівняння з іншими відповідними роботами, результати на різних етапах структури відстеження також порівнюються для більш детальної оцінки. Базові значення для порівняння коротко узагальнені наступним чином.

Спочатку порівнюються кілька методів, які використовуються на етапах виявлення та відстеження 2D м'яча. Крім того, деякі результати 3D м'яча також перевіряються в 2D.

VGG16 [5] виявляє 2D м'яч для кожної камери на основі Faster RCNN з оригінальною основою мережі VGG16.

Multi-scale VGG16 [28] виявляє м'яч на основі Faster RCNN з покращеною мережею Multi-scale VGG16.

Для чесного порівняння наступні базові значення використовують результати виявлення м'яча Multi-scale VGG16. Оскільки це покращена основна мережа порівняно з оригінальною VGG16, також коротко відносимося до неї як iVGG16 у цій роботі. Префікс "i" у iECO, iKCF і iKalman у цій роботі має той самий зміст.

KCF-based [4] виявляє м'яч за допомогою покращеної VGG16 і відстежує м'яч за допомогою відстежувача на основі KCF, який покращений таким же чином, як і ECO-based відстежувач.

ECO-based [6] покращує відстеження 2D м'яча в кожній зйомці з пропонуваним відстежувачем на основі ECO, який використовує виявлення м'яча для своєчасного оновлення моделі відстеження. Тим часом, також досліджується перехресна інформація між камерами для виправлення виявлення м'яча шляхом усунення помилкових виявлень та компенсації пропущених виявлень.

Ball-position-measurement-system [14] - це дуже пов'язана структура відстеження м'яча, створена для футболу. Спочатку виявляє м'яч за допомогою традиційного методу виявлення (віднімання переднього плану), а потім м'яч просто відстежується шляхом знаходження найближчого об'єкта від останньої позиції м'яча на кадрі, коли м'яч не виявлений. Остаточна позиція м'яча у координатах верхнього виду оцінюється шляхом інтегрування вищезгаданих результатів відстеження з різних камер. Ми реалізуємо цей метод і конвертуємо позиції м'яча в 2D точки зображення для порівняння.

iVGG16+iECO+Triangulation об'єднує кілька 2D позицій м'яча з базового відстежування ECO за допомогою техніки триангуляції. Позиції 3D проєктуються назад у кожну камеру для 2D позицій м'яча.

iVGG16+iECO+Triangulation+Kalman застосовує фільтр Калмана (просто сприймаючи м'яч як повністю лінійний рух) для згладжування траєкторії 3D м'яча, отриманої вищезгаданим базовим методом iVGG16+iECO+Triangulation. Позиції 3D також конвертуються в 2D точки зображення для оцінки.

iVGG16+iECO+Triangulation+iKalman виконується так само, як у вищезгаданому методі, за винятком того, що використовується покращений фільтр Калмана для відстеження 3D м'яча, яке ґрунтується на новому спрощеному припущенні руху для м'яча, замість повністю лінійного припущення.

Оскільки запропонована система відстеження м'яча є багатокамерною і кінцевим результатом є траєкторія м'яча в просторі 3D, було зібрано кілька

останніх алгоритмів багатокамерного відстеження м'яча як базові. Ці алгоритми коротко описані нижче.

RANSAC [27] спрямований на поліпшення траєкторії м'яча за допомогою методу RANSAC шляхом фільтрації кандидатів у м'яч, які не відповідають балістичній моделі та інтерполяції пропущених виявлень між включеннями. Цей підхід спочатку був запропонований для відстеження м'яча в наборі даних APIDIS, і він використовується як базовий. Однак цей підхід сильно покладається на доменні знання або параметри моделі, адаптовані до конкретних видів спорту, в той час як запропонований метод цього не робить.

MIP-with-constraints [12] спочатку виявляє м'яч у кожній камері на основі кольорових та рухових гістограм, а потім безпосередньо триангулює виявлення для генерації 3D координат без процедури відстеження 2D м'яча, як у запропонованій структурі. Ще одна відмінність порівняно з розробленим методом полягає в тому, що він накладає фізичні обмеження на м'яч, відповідно, коли він летить або знаходиться у власності гравцем, і нарешті формулює проблему відстеження м'яча як задачу змішаної цілочисельної програми (MIP). Однак потрібно спочатку побудувати модель взаємодії між м'ячем та гравцями, і результати відстеження м'яча серйозно залежать від якості відстеження гравців. Наприклад, запропонований метод переважно спрямований на поліпшення відстеження 2D м'яча шляхом інтеграції перехресної інформації між камерами.

Ball-position-measurement-system [14] був описаний вище, коли використовувався для оцінки 2D м'яча. Щоб порівняти його в просторі 3D, також використовуємо запропонований метод злиття 3D позицій для перетворення 2D місцезнаходжень м'яча в простір 3D.

Розроблений підхід складається з кількох версій, коли різні методи застосовуються на різних етапах запропонованої структури. Аналогічно базовим значенням 2D м'яча, VGG16 представляє виявлення 2D м'яча на основі Faster RCNN з оригінальною основою мережі VGG16, тоді як iVGG16 - це покращена Multi-scale VGG16. Так само, iKCF та iECO означають покращені трекери KCF та

ECO відповідно на етапах відстеження 2D м'яча. Триангуляція представляє запропоновану операцію злиття 3D позицій м'яча. Kalman позначає відстеження 3D м'яча за допомогою оригінального фільтра Калмана, який просто сприймає м'яч як повністю лінійний рух, тоді як iKalman означає покращене відстеження Калмана, що ґрунтується на новому спрощеному припущенні руху м'яча. Ну і використання додаткового шару CNN піднімає показники на наступний рівень точності, вищий за попередні. Кінцевий результат запропонованої структури відстеження м'яча відноситься до iVGG16+iECO+Triangulation+iKalman+CNN.

Оскільки запропонований метод спрямований на покращення точності відстеження м'яча як у 2D, так і в 3D просторі, проведено два види порівняльних експериментів для оцінки роботи багатокамерної системи відстеження м'яча. Один експеримент проводили для м'яча у кожному з 2D виглядів, а інший - для м'яча у просторі 3D. Місцезнаходження м'яча у 2D було позначене вручну, а в просторі 3D використовувалася 3D триангуляція.

У відстеженні м'яча в 3D спочатку зливаємо кілька позицій м'яча у 2D після вищезазначеного відстеження у 2D, а потім використовуємо покращений фільтр Калмана для згладжування траєкторії.

Покращення є значними збільшується з 0.79 (iVGG16 + iECO + Triangulation) до 0.92 (iVGG16 + iECO + Triangulation + iKalman) і до 0.95 (iVGG16 + iECO + Триангуляція + iKalman + CNN). Запропонований метод відстеження м'яча у 3D просторі покращує продуктивність, оскільки він інтерполює деякі пропущені м'ячі та фільтрує деякі викиди. У порівнянні з іншими версіями запропонованого методу, вони видаються відразу гіршими.

Порівнюючи запропонований метод з іншими відомими роботами, він також має великі переваги. Горизонтальна вісь - це поріг відстані у просторі 3D від 1 до 1000 см, тоді як значення точності - це відсоток кадрів, чиє оцінене місцезнаходження знаходиться в межах кожної відстані порогу. Легко помітити, що запропонований метод працює краще за декілька сучасних методів. У цьому експерименті застосували систему вимірювання позиції м'яча на наборі даних

APIDIS, скопіювавши рейтинг точності з RANSAC та MIP-with-constraints з [12]. При порозі відстані 200 см RANSAC, MIP-with-constraints система вимірювання позиції м'яча отримують рейтинги точності 0.50, 0.62 та 0.58 відповідно. У порівнянні, рейтинг точності становить 0.79 відразу після методу злиття 3D позицій та нарешті досягає 0.92 після згладжування траєкторії у 3D просторі з покращеним відстеженням Калмана і сягає 0.95 при ущільненні перебору даних для додаткового навчання через CNN.

У цій роботі пропонується новий метод на основі доповнення вже існуючих методів [4] глибоким навчанням для відстеження об'єктів при багатокамерному трекінгу у спортивних заходах. Спочатку застосовується детектор на основі CNN з мережею основного рівня VGG16 для виявлення 2D м'яча. Друге місце в роботі займає вдосконалення трекера ECO для 2D відстеження м'яча, в якому частково може бути вирішено питання заслонювання завдяки інформації між видами, і проблему розсіяння шаблону відстеження можна до певної міри полегшити, оновлюючи модель відстеження результатами виявлення м'яча. Далі послідовно вводиться метод фузії позицій 3D на основі триангуляційної техніки та підходу до відстеження 3D м'яча із поліпшеним фільтром Калмана для забезпечення плавної траєкторії 3D м'яча. Експерименти як для 2D, так і для 3D м'яча на загальнодоступному наборі даних показують, що запропонована модель відстеження м'яча є більш ефективною, ніж кілька пов'язаних робіт з суттєвими заслоняннями та навіть помилками калібрування. Внесок в основному полягає в значних поліпшеннях на різних етапах моделі методу, зокрема це виявлення 2D м'яча шляхом введення мультимасштабних характеристик, відстеження 2D м'яча за рахунок дослідження інформації між видами та своєчасного оновлення моделі відстеження результатами виявлення, а також відстеження 3D м'яча з поліпшеним фільтром Калмана на основі нового спрощеного припущення руху м'яча. Крім того, порівняно з існуючими комерційними продуктами, запропонований метод не вимагає спеціального обладнання, а лише декількох звичайних відеокамер,



тому він має невеликі витрати. Майбутня робота буде спрямована на те, як інтегрувати виявлення та відстеження м'яча в універсальну навчальну структуру.

### 3.2 Опис використаних програмних засобів

Для розробки методу детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання було використано вже існуючі методи і скомбіновано їх з технологією глибокого навчання CNN, тим самим створено новий метод, що потенційно має вищу точність визначення об'єктів, для подібних задач.

А саме мною було обрана прикладна сфера застосування багатокамерного трекінгу об'єктів в спортивних змаганнях, для відео з задачею відслідковування руху м'яча.

Для програмної реалізації цієї задачі були використані наступні програмні засоби:

**TensorFlow** рис. 3.1 є відкритою бібліотекою для чисельних обчислень, спеціалізованою на створенні та навчанні нейронних мереж. Розроблений Google Brain Team, TensorFlow став одним з найпопулярніших інструментів у сфері глибокого навчання.

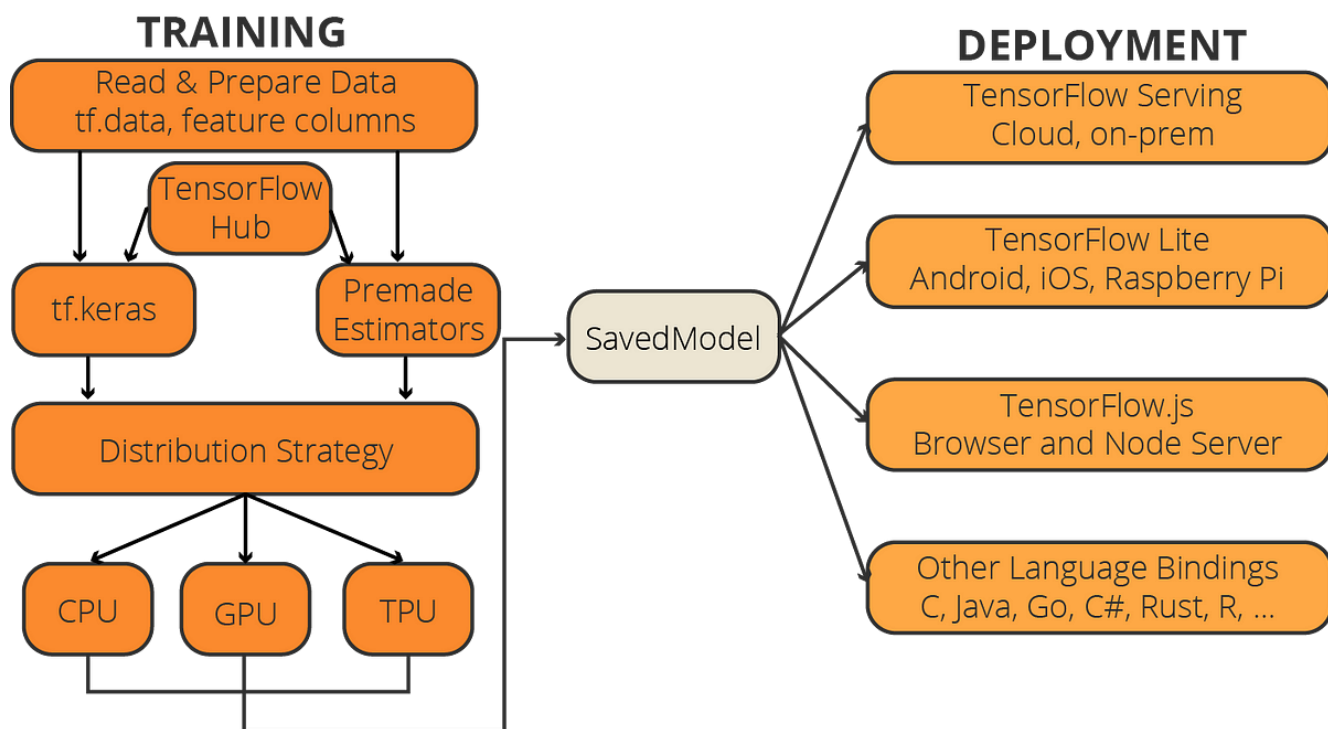


Рис. 3.1 Схема роботи з TensorFlow [9]

Основні функції та характеристики:

1. Гнучкість. TensorFlow дозволяє створювати різні типи нейронних мереж, від простих до складних архітектур.
2. Спрощена робота з нейронними мережами. TensorFlow надає високорівневий API, як Keras, що полегшує розробку нейронних мереж для початківців.
3. Масштабованість. Можна працювати як на одному комп'ютері, так і на великому кластері серверів для навчання складних моделей.
4. Візуалізація графів. TensorFlow надає інструменти для візуалізації та розуміння графів обчислень нейронних мереж.

Використання TensorFlow:

- Створення моделей. Для створення моделі з нуля, використовуючи низькорівневий API TensorFlow або вищорівневі бібліотеки, такі як

Keras.

- Навчання моделей. TensorFlow дозволяє навчати моделі на великому обсязі даних, використовуючи GPU або TPU для прискорення обчислень.
- Інтеграція з різними пристроями. З TensorFlow можна інтегрувати навчені моделі на мобільні пристрої або вбудовані системи.

Спільнота та підтримка:

TensorFlow має широку спільноту користувачів та активну базу знань. Існує безліч онлайн-ресурсів, від документації до курсів та блогів, що допомагають вирішувати проблеми та засвоювати нові концепції.

**Keras** рис. 3.2 є високорівневим API для створення, навчання та тестування нейронних мереж. Він дозволяє швидко та легко створювати моделі глибокого навчання, маючи зручний інтерфейс для початківців.

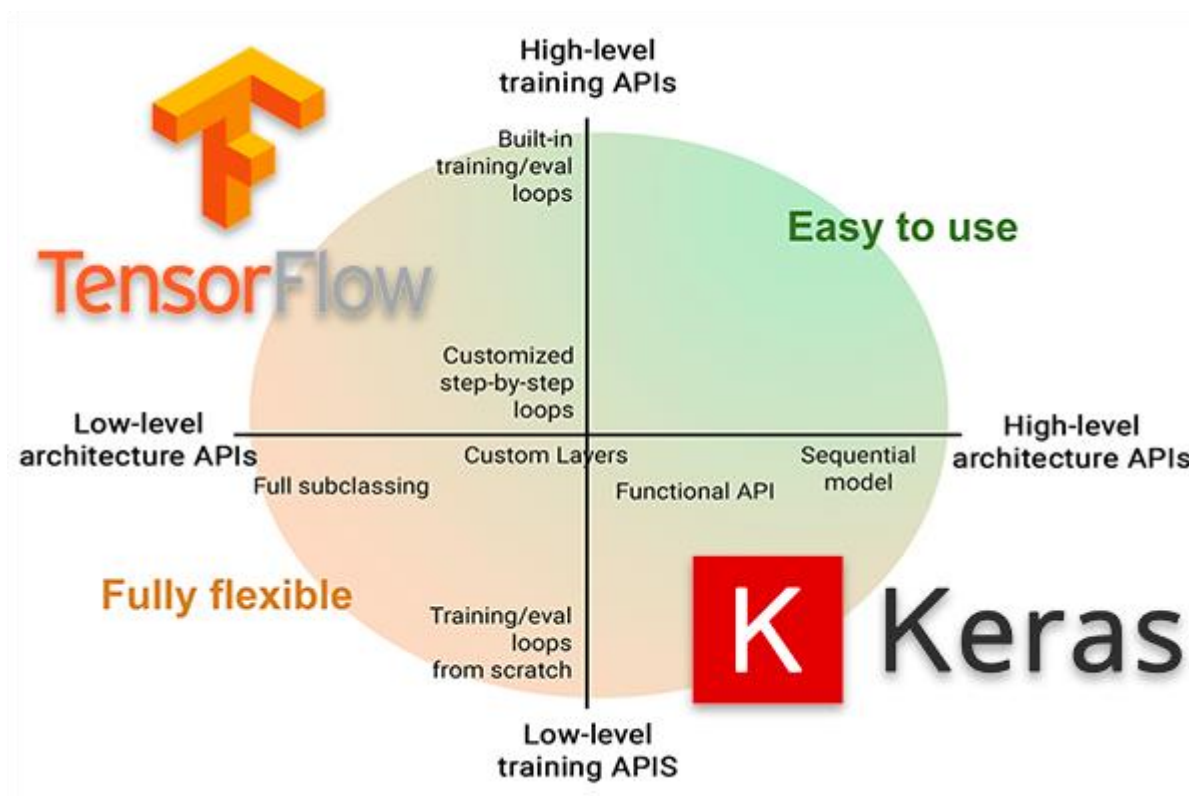


Рис. 3.2 Схеми взаємозв'язку TensorFlow та Keras [5]

Основні функції та характеристики:

1. Простота використання. Keras пропонує простий та зрозумілий для користувача синтаксис, що дозволяє швидко створювати та тестувати нейронні мережі.
2. Модульність. Модель створюється за допомогою з'єднання різних шарів, що робить його дуже гнучким.
3. Сумісність з TensorFlow та іншими бібліотеками. Keras може використовувати різні бекенди, включаючи TensorFlow, Theano та CNTK.
4. Навчання моделей. Keras пропонує зручний інтерфейс для навчання моделей, використовуючи методи оптимізації та функції втрат.

Використання Keras:

- Створення моделей. Створювати моделі, додаючи шари та конфігуруючи їх параметри.
- Навчання моделей. Використовується метод `.fit()` для навчання моделі за допомогою навчальних даних.
- Тестування та оцінка. Після навчання використовується метод `.evaluate()` для тестування та оцінки моделі на валідаційних даних.

Спільнота та підтримка:

Keras має активну спільноту користувачів, що призвело до широкої підтримки та доступної документації. Існують відкриті навчальні матеріали, форуми та блоги, які допоможуть навчатись та розвиватися у використанні цього інструменту.

**OpenCV** (Open Source Computer Vision Library) рис. 3.3 - це відкрита бібліотека комп'ютерного зору та обробки зображень, яка має широкі можливості для роботи з відео, зображеннями та комп'ютерним зором.

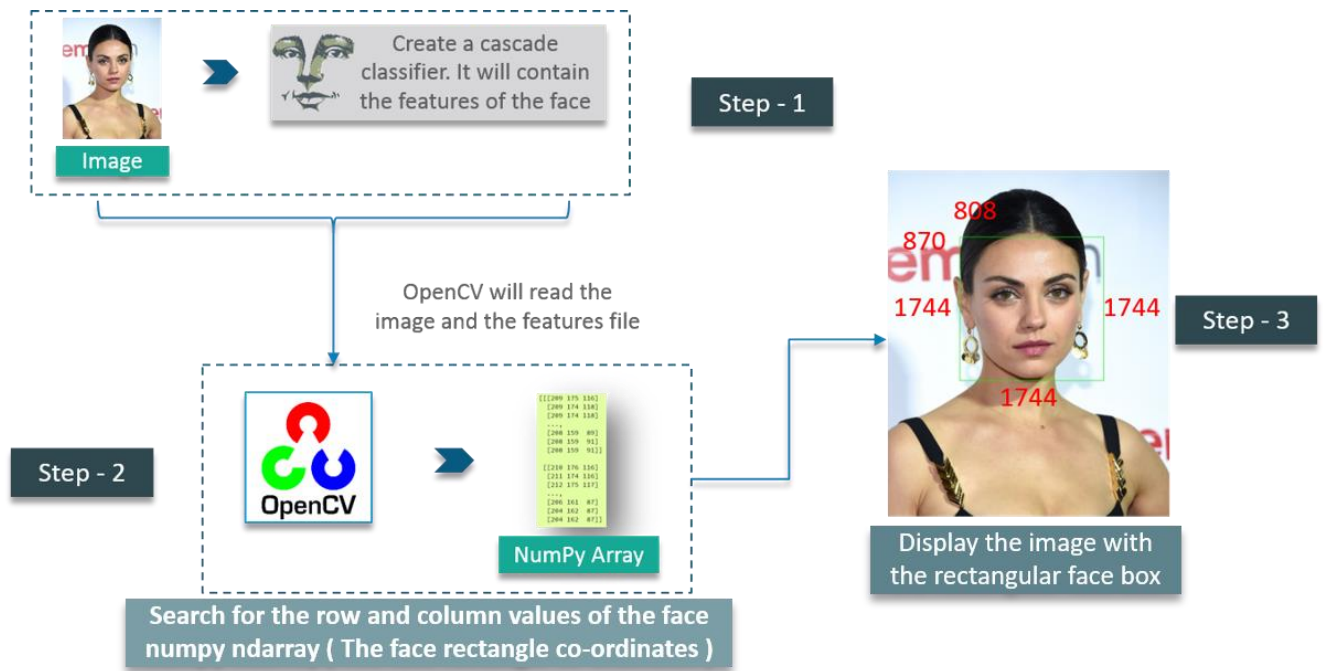


Рис. 3.3 Схема роботи з OpenCV [4]

В подальшій роботі з OpenCV виконуються наступні підходи:

#### 1. Робота з зображеннями та відео:

OpenCV надає інструменти для завантаження, зберігання та обробки зображень та відео. Це може бути все: від простого зчитування та відображення зображення до складних операцій обробки відео, таких як виявлення об'єктів, відстеження руху, витягування функцій та багато іншого.

```
import cv2
```

```
import matplotlib.pyplot as plt
```

```
# Зчитування зображення
```

```
image = cv2.imread('example_image.jpg')
```

```
# Зміна розміру зображення
```

```
resized_image = cv2.resize(image, (new_width, new_height))
```

```
# Відображення зображення
plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
plt.axis('off')
plt.show()
```

## 2. Виявлення об'єктів:

OpenCV має функції для виявлення об'єктів на зображеннях та відео за допомогою різних методів, таких як каскади Хаара, детектори ключових точок та дескриптори, нейронні мережі тощо.

# Використання каскадів Хаара для виявлення облич на зображенні

```
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
'haarcascade_frontalface_default.xml')
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray_image, scaleFactor=1.1, minNeighbors=5)
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)
```

## 3. Обробка та аналіз зображень:

OpenCV має широкий набір функцій для обробки зображень: від фільтрів та морфологічних операцій до геометричних перетворень, порівнянь зображень та витягування особливостей.

# Застосування фільтру Гаусса для згладжування зображення

```
blurred_image = cv2.GaussianBlur(image, (5, 5), 0)
```

# Порівняння двох зображень

```
difference = cv2.absdiff(image1, image2)
```

**Dlib** - це високоякісна бібліотека, яка включає інструменти для комп'ютерного зору, машинного навчання та обробки зображень. Вона має широкий спектр функцій, від розпізнавання облич рис. 3.4 до виявлення об'єктів та витягування ознак.

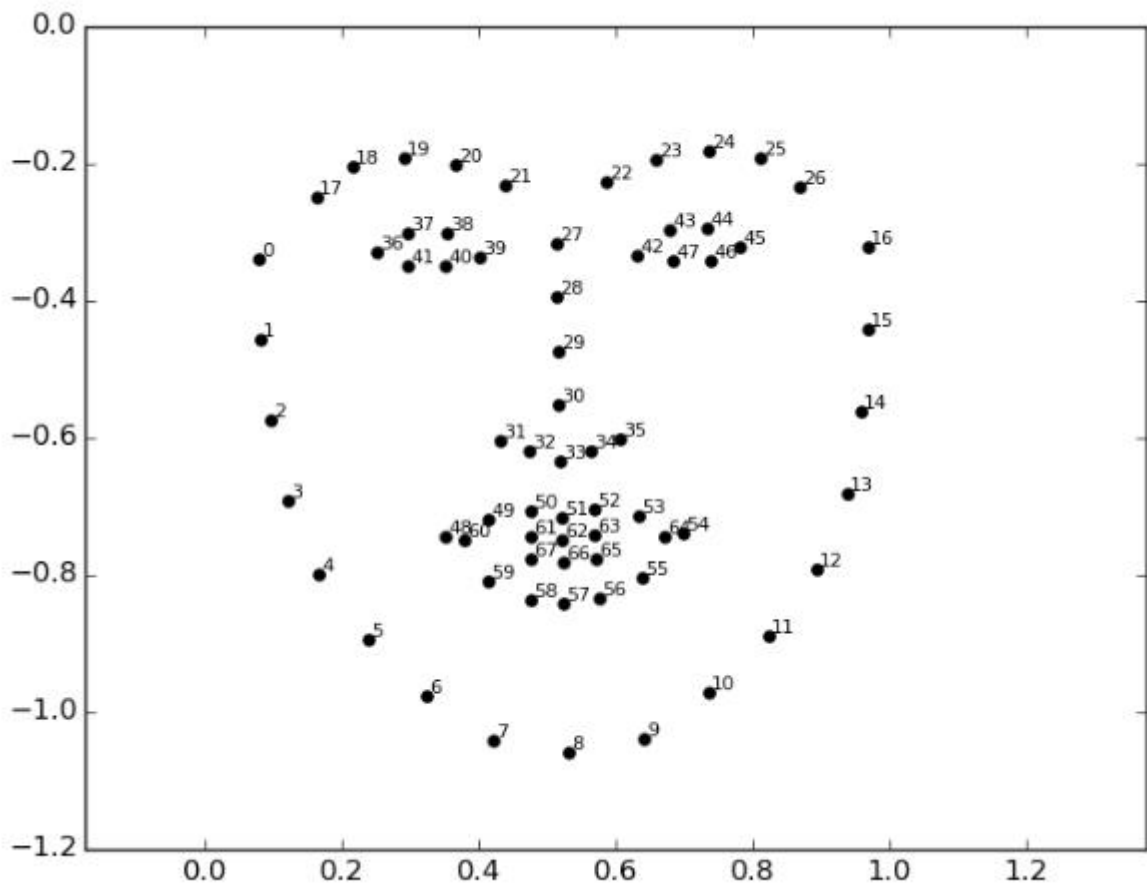


Рис. 3.4 Приклад результатів роботи Dlib [16]

В роботі застосовані принципи роботи з Dlib:

#### 1. Розпізнавання облич:

Dlib містить модуль для розпізнавання та локалізації облич на зображеннях та відео. Цей модуль може використовувати глибокі нейронні мережі для ефективного виявлення та ідентифікації облич.

```
import dlib
```

```
import cv2
```

```
# Завантаження моделі для розпізнавання облич
```

```
detector = dlib.get_frontal_face_detector()
```

```
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

```
# Завантаження зображення та виявлення облич
image = cv2.imread('face_image.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = detector(gray)
for face in faces:
    landmarks = predictor(gray, face)
# Отримання координат ключових точок обличчя та їх використання
```

## 2. Детектування об'єктів:

Dlib також має можливості для виявлення об'єктів на зображеннях, таких як автомобілі, тварини тощо.

```
# Завантаження моделі для виявлення об'єктів
detector = dlib.simple_object_detector("detector.svm")

# Виявлення об'єктів на зображенні
detections = detector(image)
for detection in detections:
    # Обробка виявлених об'єктів
```

## 3. Машинне навчання:

Dlib має інструменти для навчання моделей, включаючи підтримку SVM (метод опорних векторів) та інших алгоритмів для класифікації та регресії.

```
# Навчання класифікатора на основі SVM
trainer = dlib.svm_c_trainer_histogram_intersection()
classifier = trainer.train(features, labels)
```

**NumPy** - це потужна бібліотека для обчислень рис. 3.5 у Python, що надає підтримку для роботи з багатовимірними масивами та математичними функціями.



# Uses of NumPy

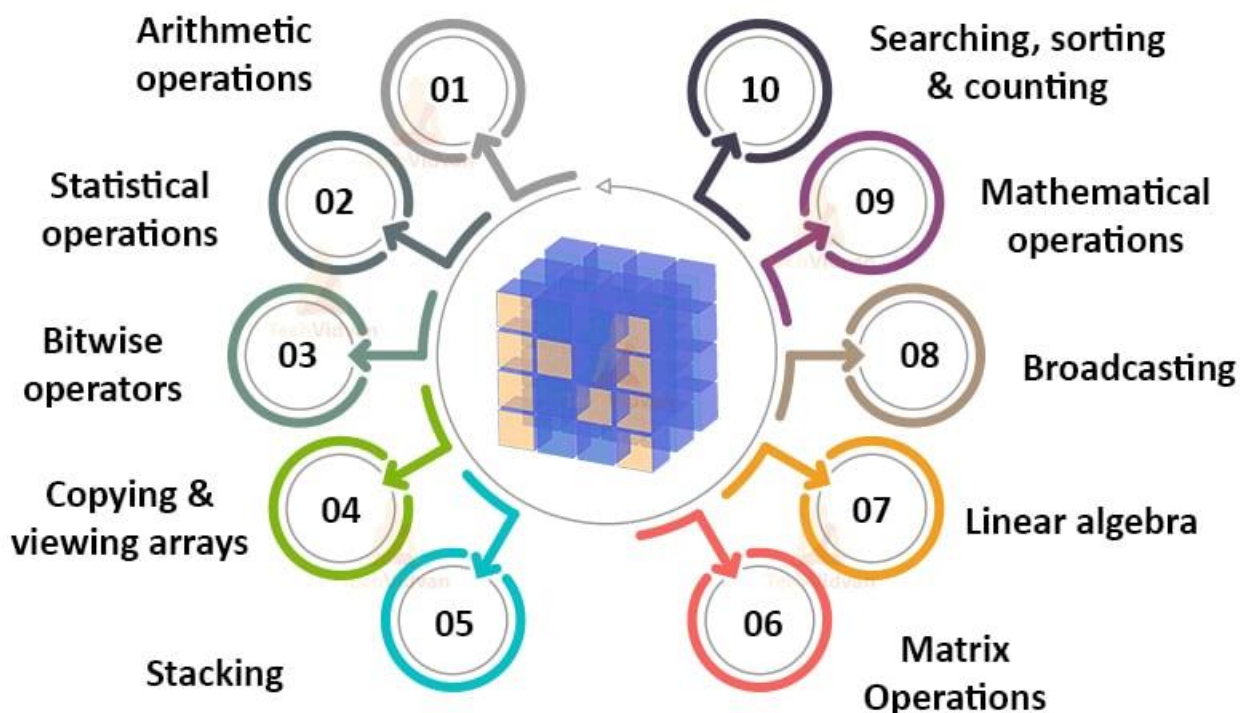


Рис. 3.5 Використання NumPy [4]

В проєкті використовуються наступні підходи до NumPy :

## 1. Робота з масивами:

NumPy надає об'єкт `ndarray`, який дозволяє ефективно працювати з багатовимірними масивами даних.

```
import numpy as np
```

```
# Створення масиву NumPy
arr = np.array([[1, 2, 3], [4, 5, 6]])
```

```
# Операції з масивами
arr_sum = np.sum(arr)
arr_mean = np.mean(arr)
arr_transpose = arr.T
```

## 2. Математичні операції:

NumPy має багато математичних функцій для роботи з масивами, включаючи операції лінійної алгебри, тригонометричні функції, функції порівняння та інші.

# Лінійна алгебра

```
dot_product = np.dot(arr1, arr2)
```

```
inverse_matrix = np.linalg.inv(matrix)
```

# Тригонометричні функції

```
sin_values = np.sin(arr)
```

```
cos_values = np.cos(arr)
```

## 3. Індексція та зрізи:

NumPy дозволяє здійснювати ефективну індексцію та зрізи масивів для отримання підмасивів або окремих значень.

# Індексція та зрізи

```
subarray = arr[1, 1:3]
```

```
masking = arr[arr > 3]
```

Використання:

- **Обробка даних.** Використання NumPy для маніпулювання даними в процесі підготовки для навчання моделей глибокого навчання.
- **Математичні обчислення.** Застосування NumPy для обчислення статистичних параметрів або виконання операцій лінійної алгебри.
- **Зображення та обробка відео.** Можливість NumPy працювати з матрицями дозволяє ефективно обробляти зображення та відео для виявлення облич, обробки кадрів тощо.

NumPy є дуже корисним у даному дипломному проекті для обробки даних, математичних обчислень та роботи зі зображеннями або відео.

**SciPy** - це бібліотека для наукових обчислень рис. 3.6 та технічних обчислень у Python. Вона базується на NumPy та має додаткові можливості для оптимізації, інтерполяції, обробки сигналів, статистики та інших обчислень.

```
import matplotlib.pyplot as plt
plt.plot(x, f(x))
plt.scatter(start, f(start))
plt.scatter(optimized, f(optimized))
plt.legend(['Function -sin(x)', 'Starting point', 'Optimized minimum']);
```

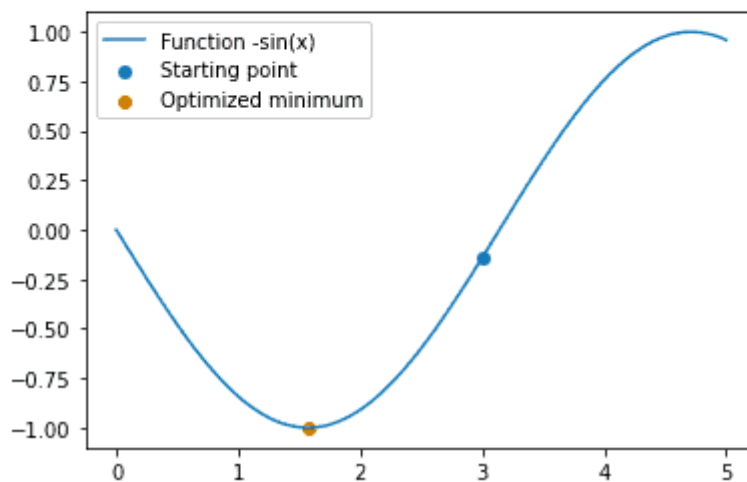


Рис. 3.6 Приклад роботи з SciPy [5]

Використання в проекті SciPy:

#### 1. Оптимізація:

SciPy має функції для оптимізації, включаючи методи пошуку мінімуму/максимуму, оптимізацію функцій та розв'язання оптимізаційних завдань.

```
from scipy.optimize import minimize
```

```
# Мінімізація функції
```

```
def func(x):
```

```
    return x**2 + 5*np.sin(x)
```

```
result = minimize(func, x0=0)
```

## 2. Інтерполяція:

SciPy надає інструменти для інтерполяції даних, що дозволяє побудувати гладку функцію на основі обмеженого набору точок.

```
from scipy.interpolate import interp1d
```

```
# Інтерполяція даних
```

```
x = np.linspace(0, 10, 10)
```

```
y = np.sin(x)
```

```
f = interp1d(x, y, kind='cubic')
```

## 3. Обробка сигналів та зображень:

SciPy має інструменти для обробки сигналів, фільтрації, просторової обробки зображень та роботи з звуком.

```
from scipy import signal, ndimage
```

```
# Фільтрація сигналів
```

```
filtered_signal = signal.medfilt(data)
```

```
# Обробка зображень
```

```
blurred_image = ndimage.gaussian_filter(image, sigma=2)
```

## Використання SciPy в проекті:

- Обробка даних та оптимізація моделей. Використання SciPy для оптимізації параметрів моделей, розв'язання задач оптимізації та обробки даних.
- Інтерполяція та апроксимація. Застосування інтерполяції для роботи з обмеженими даними, що може бути корисним у відновленні сигналів або обробці даних.
- Обробка зображень та сигналів. Використання функцій SciPy для

фільтрації, обробки зображень та сигналів, що може бути корисним у вашому проєкті для аналізу даних.

SciPy є потужним інструментом для розв'язання різноманітних завдань у наукових обчисленнях, оптимізації та обробці даних у моєму дипломному проєкті.

**Matplotlib** - це бібліотека для створення візуалізацій та графіків рис. 3.7 у Python. Вона дозволяє побудувати різноманітні типи графіків, діаграм, гістограм, 3D-графіків тощо.

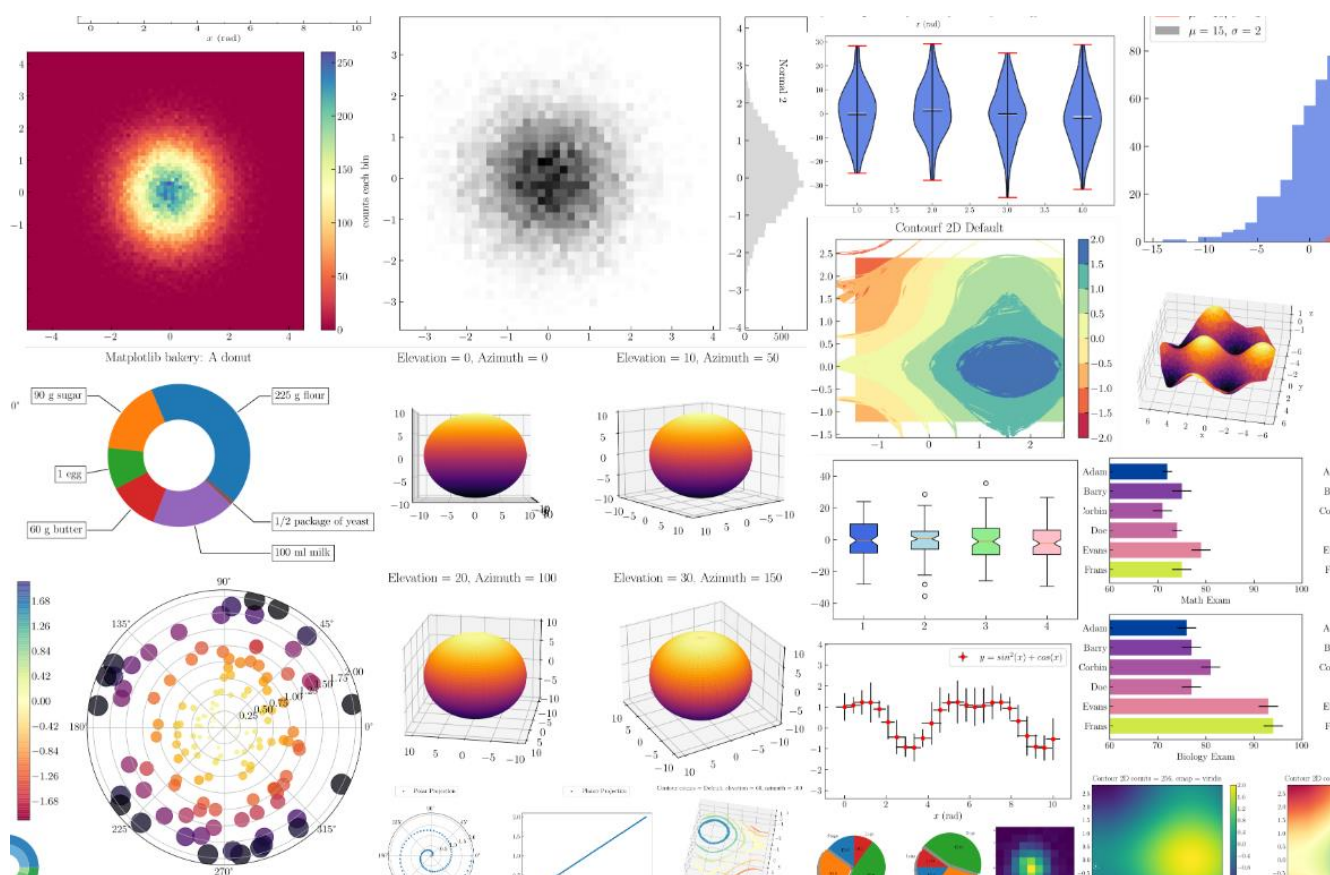


Рис. 3.7 Приклади роботи з візуалізацією Matplotlib [12]

Matplotlib використовувалось наступним чином:

1. Побудова лінійних та точкових графіків:

Matplotlib дозволяє побудувати лінійні графіки для відображення залежностей між даними, а також точкові графіки для відображення окремих точок.

```
import matplotlib.pyplot as plt
```

```
# Побудова лінійного графіка
```

```
plt.plot(x, y, label='sin(x)')
```

```
plt.xlabel('x')
```

```
plt.ylabel('sin(x)')
```

```
plt.legend()
```

```
plt.show()
```

## 2. Створення гістограм та діаграм:

Matplotlib надає можливість побудувати гістограми для відображення розподілу даних або діаграми для відображення категорійних даних.

```
# Побудова гістограми
```

```
plt.hist(data, bins=10)
```

```
plt.xlabel('Value')
```

```
plt.ylabel('Frequency')
```

```
plt.show()
```

## 3. Візуалізація даних у 3D:

Matplotlib також має можливість створювати 3D-графіки для відображення тривимірних даних або поверхонь.

```
from mpl_toolkits.mplot3d import Axes3D
```

```
# 3D-візуалізація даних
```

```
fig = plt.figure()
```

```
ax = fig.add_subplot(111, projection='3d')
```

```
ax.scatter(xs, ys, zs, c='r', marker='o')
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
plt.show()
```

Використання Matplotlib в проекті:

- Візуалізація даних. Matplotlib може бути використаний для створення графіків, які демонструють розподіл даних, залежності та тренди.
- Представлення результатів. Використання графіків Matplotlib для відображення результатів досліджень, що допомагає в легшому розумінні та аналізі.
- Візуалізація моделей. Використання графіків для відображення внутрішньої структури моделей або результатів прогнозування.

Matplotlib - це потужний інструмент для візуалізації даних, що може стати важливим компонентом мого проекту для представлення результатів та відображення залежностей між даними.

**Jupyter Notebook** - це інтерактивне середовище рис. 3.8 для програмування, яке дозволяє об'єднувати код, тексти, візуалізації та пояснення у документованій формі. Детальний огляд функціональностей та можливостей Jupyter Notebook:

1. Експерименти з кодом – Ви можете проводити експерименти з різними алгоритмами та варіантами реалізації коду, зберігаючи результати кожного кроку у вигляді окремих комірок.
2. Візуалізація даних – Використовуйте Jupyter для створення графіків та візуалізацій для представлення даних у зрозумілій формі.
3. Документація та пояснення – Використовуйте текстові комірки для пояснень та документації до вашого коду, щоб зрозуміло пояснити його функціонал.

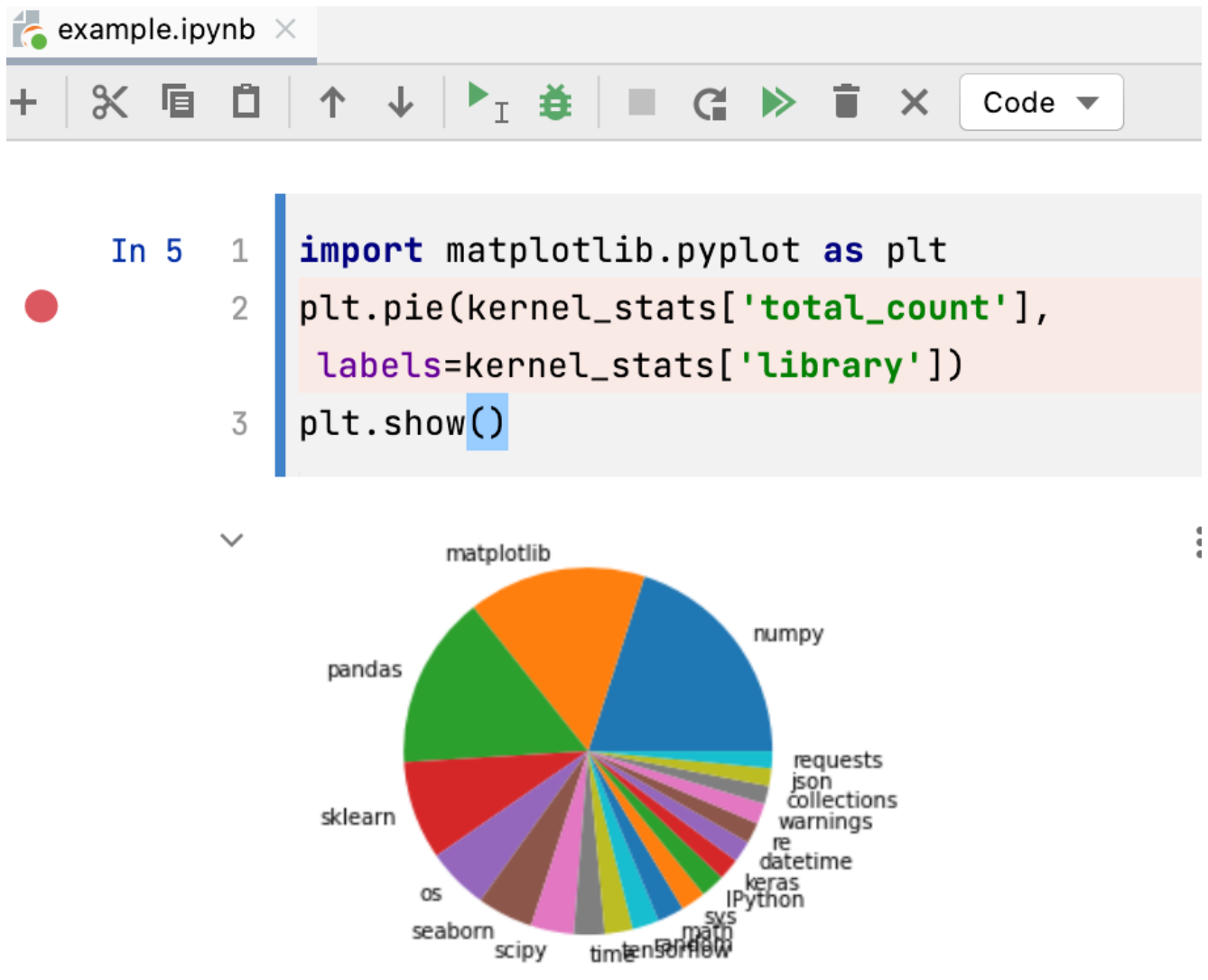


Рис. 3.8 Приклад роботи в Jupyter Notebook [6]

**PyCharm** рис. 3.9 - це інтегроване середовище розробки (IDE) для Python, розроблене компанією JetBrains. Воно надає широкий набір інструментів для розробки, налагодження та управління проектами Python. Детальніше про можливості PyCharm та його корисність для даного дипломного проекту:

- PyCharm надає розширені можливості для написання та редагування коду Python, з підтримкою автодоповнення, рефакторингу, перевірки синтаксису та багато іншого.
- Ви можете використовувати PyCharm для налагодження свого коду, встановлюючи точки зупинки, відстежуючи значення змінних,



перевіряючи виклики функцій тощо.

- Управління проектами. PyCharm дозволяє легко створювати та керувати проектами Python, включаючи підтримку систем керування версіями, віртуальних середовищ та інші інструменти для організації вашого коду.
- Підтримка графіків та візуалізацій у PyCharm дозволяє вам швидко переглядати та аналізувати дані, а також відлагоджувати свій код.

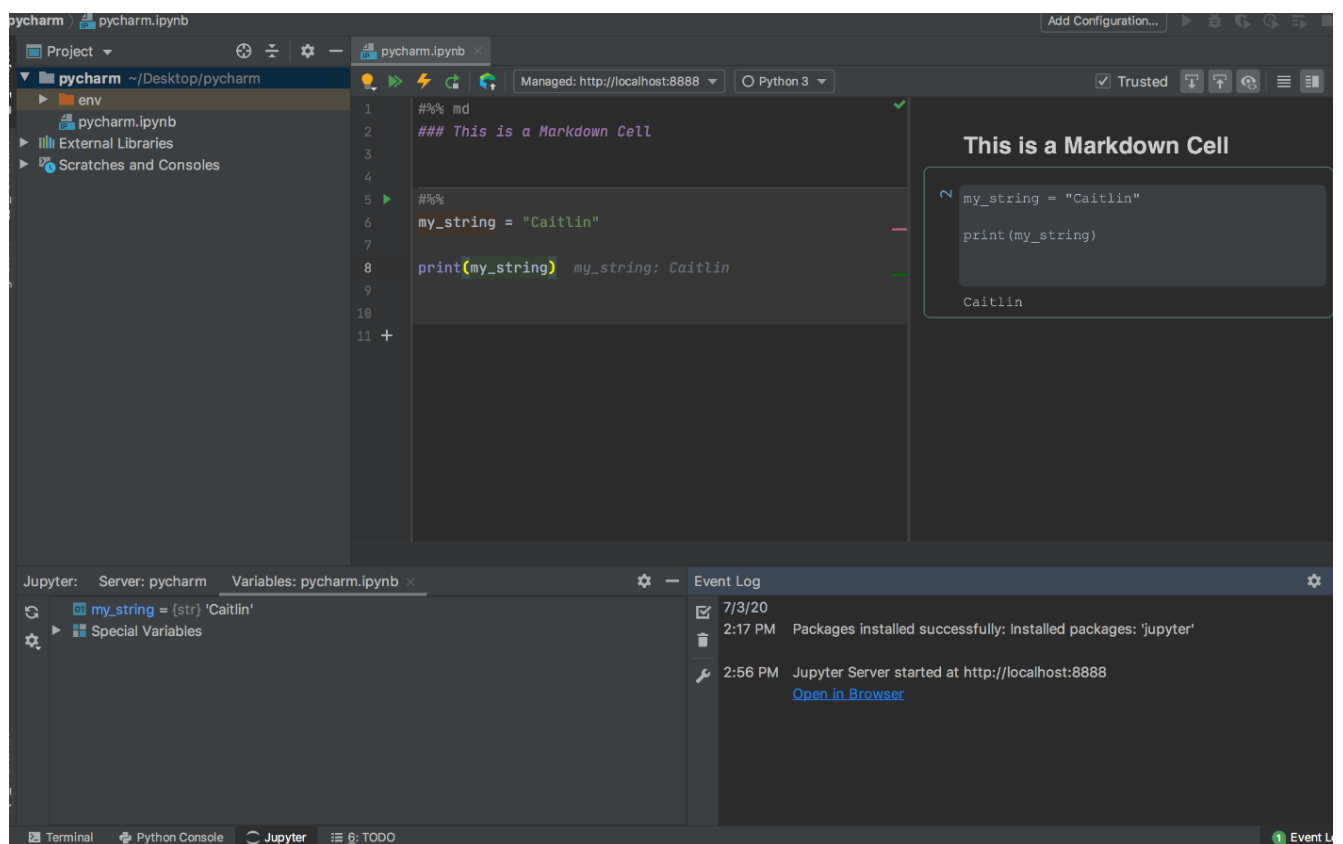


Рис. 3.9 Приклад роботи в PyCharm

### 3.3 Опис структури проекту

Структура даного проекту для розробки методу детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання включає наступні складові:

### 1. Директорія "Data":

- Raw\_Data – Тут зберігається вихідна та оброблена вхідна інформація. До неї входять відеофайли, зображення та інші дані.

Приклад:

- video\_1.mp4
- video\_2.mp4
- image\_1.jpg
- ...

- Processed\_Data – Ця директорія містить підготовлені дані для моделей, вже готові до використання в навчанні або тестуванні моделей.

Приклад:

- preprocessed\_data.csv
- normalized\_images/
  - image\_1\_normalized.jpg
  - image\_2\_normalized.jpg
- ...

### 2. Директорія "Models":

- CNN\_Models – Тут знаходяться файли моделей глибокого навчання (Convolutional Neural Networks). Кожен файл може представляти окрему модель або її варіант.

Приклад:

- cnn\_model\_v1.h5
- cnn\_model\_v2.h5
- cnn\_model\_v3.h5

- Tracking\_Models – Директорія для файлів, пов'язаних із трекінгом об'єктів та моделями відстеження.

Приклад:

- tracking\_model\_1.pkl
- tracking\_model\_2.pkl

### 3. Директорія "Notebooks":

- `Data_Preprocessing.ipynb` – записник для попередньої обробки даних перед навчанням моделей.
- `Model_Training.ipynb` – записник для навчання моделей глибокого навчання на підготовлених даних.
- `Evaluation_and_Analysis.ipynb` – записник для оцінки результатів моделей та аналізу їх продуктивності.

### 4. Директорія "Utils":

- `Data_Processing_Utils.py` – містить функції для попередньої обробки та чистки даних.

Приклад:

```
def resize_image(image, dimensions):
```

```
    # код для зміни розмірів зображення до заданих dimensions
```

```
    pass
```

```
def remove_noise(data):
```

```
    # код для видалення шуму з даних
```

```
    pass
```

- `Model_Utils.py` – утиліти для навчання моделей, включаючи функції побудови та оцінки моделей.

Приклад:

```
def build_model(layers, activation):
```

```
    # код для побудови моделі з заданими параметрами
```

```
    pass
```

```
def evaluate_model(model, test_data):
```

```
    # код для оцінки точності моделі на тестових даних
```

```
    pass
```

### 5. Директорія "Reports":

- Project\_Report.docx – документ, який містить опис проекту, методологію, результати, висновки та іншу важливу інформацію.
- Presentations – тут можуть знаходитися презентації, які використовуються для захисту проекту.

Приклад:

- presentation\_1.pptx

- presentation\_2.pdf

#### 6. Файл "requirements.txt":

Файл, що містить перелік всіх бібліотек та залежностей, необхідних для виконання проекту.

Приклад:

numpy==1.21.3

pandas==1.3.4

tensorflow==2.7.0

opencv-python==4.5.3

#### 7. Файл "README.md":

Короткий опис проекту, інструкції по запуску, посилання на документацію та інше.

Приклад:

# Project Title

Short description of the project.

## Installation

Instructions on how to install necessary dependencies.

## Usage

Instructions on how to use the project.

## References

Any relevant references or resources.

Це загальна структура, яка може змінюватися відповідно до специфіки потреб. Важливо підтримувати належну організацію та документацію для кращого розуміння та розвитку проекту.

### 3.4 Опис інтерфейсу

1. Домашня сторінка проекту. Основне меню:

- Data Management: Перегляд та управління вхідними та обробленими даними.
- Model Training: Навчання та оцінка моделей глибокого навчання.
- Analysis & Visualization: Аналіз результатів та візуалізація даних.
- Reports & Presentations: Доступ до звітів, презентацій та інших документів.

2. Data Management. Варіанти обробки даних:

- Завантаження вхідних даних (відео, зображення).
- Попередня обробка даних (очистка, нормалізація, зміна розміру).
- Відображення та видалення шуму з даних.

3. Model Training. Навчання та оцінка моделей:

- Вибір типу моделі для навчання (CNN, LSTM, тощо).
- Налаштування параметрів моделі (шари, функції активації, оптимізатори).
- Завантаження попередньо навчених моделей.
- Оцінка точності та продуктивності моделей.

4. Analysis & Visualization. Аналіз результатів та візуалізація:

- Графіки процесу навчання моделі.
- Відображення результатів відстеження об'єктів на відео або зображеннях.

Статистичний аналіз точності та візуалізація результатів рис. 3.10.

## 5. Reports & Presentations. Документи та презентації:

- Завантаження та перегляд звітів проекту.
- Доступ до презентацій для презентації результатів.

## 6. Інше. Додаткові функції:

- Панель керування користувачами.
- Збереження параметрів та налаштувань проекту.

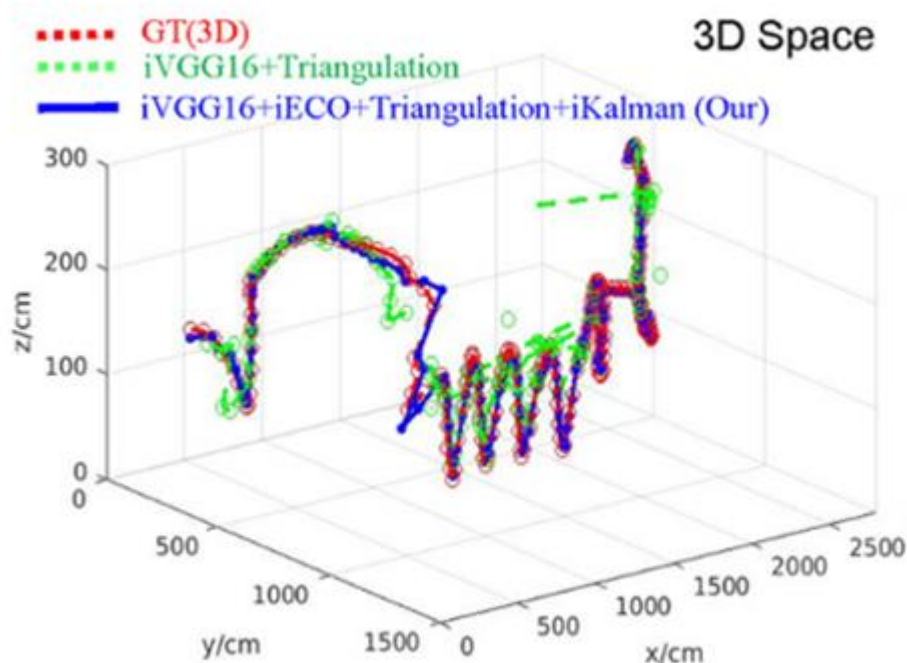


Рис. 3.10 Результат моделювання методів в 3D просторі [20]

## 3.5 Опис розроблених класів

### 1. Клас "DataProcessor":

Опис:

Цей клас забезпечує обробку та попередню підготовку даних для використання у моделях глибокого навчання. Він використовується для завантаження, обробки та очистки даних перед їх подальшим використанням.

Методи:

`load_data(data_path)` – Метод для завантаження вхідних даних з файлової системи або бази даних.

`preprocess_data(data)` – Метод для попередньої обробки та нормалізації даних перед використанням у моделях.

`remove_noise(data)` – Метод для видалення шуму або непотрібних артефактів з даних.

## 2. Клас "DeepLearningModel":

Опис:

Цей клас містить реалізації моделей глибокого навчання для детектування і трекінгу об'єктів. Він відповідає за побудову, навчання та оцінку моделей, використовуючи зазначені параметри.

Методи:

`build_model(layers, activation)` – Метод для побудови моделі з заданими шарами та функціями активації.

`train_model(train_data, validation_data)` – Метод для навчання моделі на навчальних даних.

`evaluate_model(test_data)` – Метод для оцінки точності та ефективності моделі на тестових даних.

## 3. Клас "ObjectTracker":

Опис:

Цей клас відповідає за відстеження об'єктів на основі результатів моделей глибокого навчання. Він надає засоби відстеження та візуалізації об'єктів у вхідних даних.

Методи:

`track_objects(data, model_output)` – Метод для відстеження об'єктів на вхідних даних з використанням результатів моделей.

`visualize_tracking_results(tracking_data)` – Метод для візуалізації результатів відстеження об'єктів на вхідних даних.

#### 4. Клас "DataAnalyzer":

##### Опис:

Цей клас забезпечує аналіз та візуалізацію отриманих даних та результатів моделей. Він використовується для статистичного аналізу та відображення інформації про дані та ефективність моделей.

##### Методи:

`analyze_results(results)` – Метод для аналізу та відображення результатів роботи моделей.

`visualize_data(data)` – Метод для візуалізації оброблених даних та результатів аналізу.



## ВИСНОВКИ

1. Проаналізовано існуючі підходи до багатокамерного відслідковування об'єктів. Виявлено, що за комбінування існуючих підходів виходить покращити точність відслідковування об'єктів, завдяки використанню сильних сторін різних методів.
2. Проаналізовано приклади застосування методів багатокамерного трекінгу разом з методами машинного навчання. Такі комбінації показують свою широку ефективність в різних сферах: безпеки, бізнесу, медицини, спорту тощо.
3. Вибрано приклади з гарною потенційною сумісністю з технологією глибокого навчання. А саме застосування глибокого навчання для рішень задач відслідковування м'яча в тривимірному просторі виправдало очікування і показало кращі результати точності 95% у порівнянні з методами, які використовувались раніше.
4. Вдосконалено підхід до багатокамерного відслідковування об'єктів завдяки використанню методу машинного навчання. Доповнення комбінації методів технологією згорткової нейронної мережі є модифікуванням попередньої моделі з точністю 92%.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Bobkov A. V. Multi-camera object tracking system [Electronic resource] / A. V. Bobkov, G. V. Tedeev // Engineering journal: science and innovation. – 2021. – No. 10 (118). – Mode of access: <https://doi.org/10.18698/2308-6033-2021-10-2123> (date of access: 11.12.2023). – Title from screen.
2. Deep J. Programming : 4 books in 1: python programming and crash course, machine learning for beginners, python machine learning / James Deep. – [S. l.] : Independently Published, 2020. – 502 p.
3. Deep J. Python machine learning: a hands-on beginner's guide to effectively understand artificial neural networks and machine learning using python / James Deep. – [S. l.] : Independently Published, 2019.
4. Faster R-CNN: towards real-time object detection with region proposal networks [Electronic resource] / Shaoqing Ren [et al.] // IEEE transactions on pattern analysis and machine intelligence. – 2017. – Vol. 39, no. 6. – P. 1137–1149. – Mode of access: <https://doi.org/10.1109/tpami.2016.2577031> (date of access: 11.12.2023). – Title from screen.
5. Gacovski Z. Deep learning algorithms / Zoran Gacovski. – [S. l.] : Arcler Education Inc, 2021.
6. Gupta J. Deep learning (CNN) and transfer learning: a review [Electronic resource] / Jaya Gupta, Sunil Pathak, Gireesh Kumar // Journal of physics: conference series. – 2022. – Vol. 2273, no. 1. – P. 012029. – Mode of access: <https://doi.org/10.1088/1742-6596/2273/1/012029> (date of access: 11.12.2023). – Title from screen.
7. Kantardzic M. Deep learning applications / Mehmed Kantardzic, Moamar Sayed-Mouchaweh, M. Arif Wani. – [S. l.] : Springer, 2020. – 188 p.
8. Lee S.-Y. OpenCV-based object tracking system [Electronic resource] / Sang-Young Lee // Asia-pacific journal of multimedia services convergent with art, humanities, and sociology. – 2016. – Vol. 6, no. 5. – P. 29–37. – Mode of

- access: <https://doi.org/10.14257/ajmahs.2016.05.37> (date of access: 11.12.2023). – Title from screen.
9. Marron J. S. Object oriented data analysis / James Stephen Marron, I. L. Dryden. – [S. l.] : Taylor & Francis Group, 2021.
  10. Object detection in 20 years: a survey [Electronic resource] / Zhengxia Zou [et al.] // Proceedings of the IEEE. – 2023. – P. 1–20. – Mode of access: <https://doi.org/10.1109/jproc.2023.3238524> (date of access: 11.12.2023). – Title from screen.
  11. Object tracking [Electronic resource] / ed. By H. Goszczynska. – [S. l.] : InTech, 2011. – Mode of access: <https://doi.org/10.5772/614> (date of access: 11.12.2023). – Title from screen.
  12. Object tracking technology [Electronic resource] / ed. By A. Kumar [et al.]. – Singapore : Springer Nature Singapore, 2023. – Mode of access: <https://doi.org/10.1007/978-981-99-3288-7> (date of access: 11.12.2023). – Title from screen.
  13. Ramakrishnan S. Visual object tracking with deep neural networks / Srinivasan Ramakrishnan, Paolo Spagnolo, Pier Luigi Mazzeo. – [S. l.] : IntechOpen, 2019.
  14. Rao C. R. Deep learning / C. R. Rao, Arni S. R. Srinivasa Rao, Venu Govindaraju. – [S. l.] : Elsevier Science & Technology, 2023.
  15. Roth D. E. Real-time multi-object tracking / Daniel Eugen Roth. – Konstanz : Hartung-Gorre, 2010. – 102 p.
  16. Shahriar K. M. F. Object based augmented reality and its tracking systems / Kazi Muhammad Fahim Shahriar. – [S. l.] : Independently Published, 2018.
  17. Singh K. D. Machine learning in robotics with fog/cloud computing and iot [Electronic resource] / Kiran Deep Singh, Prabh Deep Singh // EAI endorsed transactions on AI and robotics. – 2023. – Vol. 2. – Mode of access: <https://doi.org/10.4108/airo.3621> (date of access: 11.12.2023). – Title from screen.

18. Streit R. Analytic combinatorics for multiple object tracking / Roy Streit, Robert Blair Angle, Murat Efe. – [S. l.] : Springer International Publishing AG, 2021.
19. Video object tracking [Electronic resource] / Ning Xu [et al.]. – Cham : Springer Nature Switzerland, 2024. – Mode of access: <https://doi.org/10.1007/978-3-031-44660-3> (date of access: 11.12.2023). – Title from screen.
20. Visual object tracking from correlation filter to deep learning / Jun Wang [et al.]. – [S. l.] : Springer, 2022.
21. 3D object detection based on synthetic RGB image [Electronic resource] / Juntong Yun [et al.] // International journal of wireless and mobile computing. – 2021. – Vol. 20, no. 1. – P. 70. – Mode of access: <https://doi.org/10.1504/ijwmc.2021.10035668> (date of access: 19.12.2023). – Title from screen.
22. A survey of detection-based video multi-object tracking [Electronic resource] / Yan Dai [et al.] // Displays. – 2022. – P. 102317. – Mode of access: <https://doi.org/10.1016/j.displa.2022.102317> (date of access: 19.12.2023). – Title from screen.
23. A survey of video based action recognition in sports [Electronic resource] / Nur Azmina Rahmad [et al.] // Indonesian journal of electrical engineering and computer science. – 2018. – Vol. 11, no. 3. – P. 987. – Mode of access: <https://doi.org/10.11591/ijeecs.v11.i3.pp987-993> (date of access: 19.12.2023). – Title from screen.
24. Distributed multi-camera multi-target association for real-time tracking [Electronic resource] / Senquan Yang [et al.] // Scientific reports. – 2022. – Vol. 12, no. 1. – Mode of access: <https://doi.org/10.1038/s41598-022-15000-4> (date of access: 19.12.2023). – Title from screen.
25. Fei L. Multi-Object multi-camera tracking based on deep learning for intelligent transportation: a review [Electronic resource] / Lunlin Fei, Bing

- Han // *Sensors*. – 2023. – Vol. 23, no. 8. – P. 3852. – Mode of access: <https://doi.org/10.3390/s23083852> (date of access: 19.12.2023). – Title from screen.
26. Gu P. Object detection combining CNN and adaptive color prior features [Electronic resource] / Peng Gu, Xiaosong Lan, Shuxiao Li // *Sensors*. – 2021. – Vol. 21, no. 8. – P. 2796. – Mode of access: <https://doi.org/10.3390/s21082796> (date of access: 19.12.2023). – Title from screen.
27. Mittal P. Deep learning-based object detection in low-altitude UAV datasets: a survey [Electronic resource] / Payal Mittal, Raman Singh, Akashdeep Sharma // *Image and vision computing*. – 2020. – Vol. 104. – P. 104046. – Mode of access: <https://doi.org/10.1016/j.imavis.2020.104046> (date of access: 19.12.2023). – Title from screen.
28. Mooncheol Won. Air-to-fuel ratio control of spark ignition engines using Gaussian network sliding control [Electronic resource] / Mooncheol Won, S. B. Choi, J. K. Hedrick // *IEEE transactions on control systems technology*. – 1998. – Vol. 6, no. 5. – P. 678–687. – Mode of access: <https://doi.org/10.1109/87.709504> (date of access: 19.12.2023). – Title from screen.
29. Multitarget vehicle tracking and motion state estimation using a novel driving environment perception system of intelligent vehicles [Electronic resource] / Yuren Chen [et al.] // *Journal of advanced transportation*. – 2021. – Vol. 2021. – P. 1–16. – Mode of access: <https://doi.org/10.1155/2021/6251399> (date of access: 19.12.2023). – Title from screen.
30. Sullivan J. Deep learning based analysis of microstructured materials for thermal radiation control [Electronic resource] / Jonathan Sullivan, Arman Mirhashemi, Jaeho Lee // *Scientific reports*. – 2022. – Vol. 12, no. 1. – Mode of access: <https://doi.org/10.1038/s41598-022-13832-8> (date of access: 19.12.2023). – Title from screen.

# ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

## (Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-  
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### МАГІСТЕРСЬКА РОБОТА «РОЗРОБКА МЕТОДУ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ ПРИ БАГАТОКАМЕРНОМУ ТРЕКІНГУ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ»

Виконав: студент групи ПДМ-64 Буряк Роман Володимирович

Керівник: к.т.н., доц., доцент кафедри ІПЗ Золотухіна Оксана Анатоліївна

Київ - 2024

#### МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

**Мета роботи:** підвищення точності визначення положення об'єкта в тривимірному просторі при багатоканальному трекуванні за рахунок використання методів глибокого навчання.

**Об'єкт дослідження:** процес детектування об'єктів в тривимірному просторі.

**Предмет дослідження:** метод детектування об'єктів в тривимірному просторі на основі глибокого навчання.

### МЕТОДИ ТРЕКІНГУ ОБ'ЄКТІВ

Методи	Опис	Переваги	Недоліки
Методи відслідковування ознак	Використання характеристик об'єктів для їхнього трекінгу	Ефективність у визначенні особливостей об'єктів	При змінах зовнішніх ознак об'єкта втрачає його з видимості
Методи оптичного потоку	Аналіз руху об'єктів на основі піксельних змін між кадрами	Висока швидкість виявлення руху, придатність для об'єктів, що швидко рухаються	Зміни в світлі, тіні, шуми призводять до не розпізнавання об'єкта
Методи глибинного відчуття	Використання інформації про глибину для відслідковування об'єктів	Ефективність у розрізненні об'єктів у тривимірному просторі	Фізичні обмеження камер дають багато помилок при складних умовах зйомки чи великих відстанях
Методи глибокого навчання	Використання глибоких нейронних мереж для відстеження об'єктів	Ефективність у різних умовах, можливість відслідковувати складні об'єкти	Мають потреби в великій кількості даних та в великих обчислювальних витратах

3

### МАТЕМАТИЧНА ОЦІНКА ТОЧНОСТІ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ ДЛЯ ГЛИБОКОГО НАВЧАННЯ

$$\text{Точність} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%, \quad (1)$$

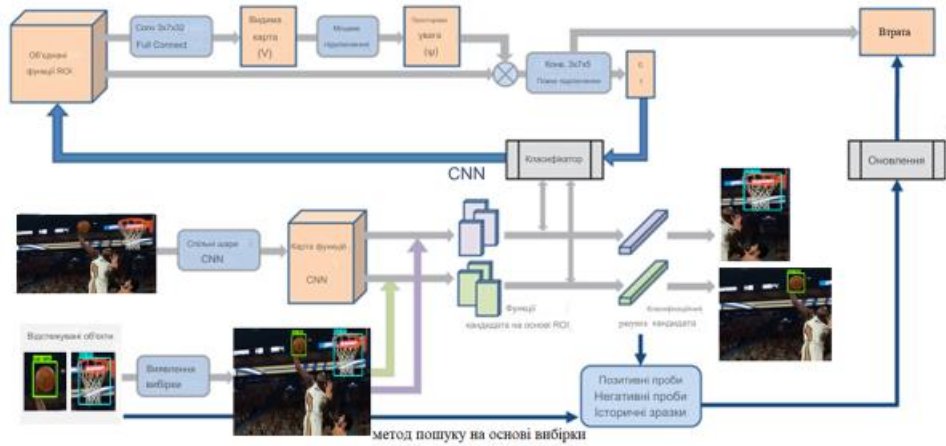
- де TP (True Positives) - кількість правильно передбачених позитивних класів (об'єкти вірно відслідковані),  
 TN (True Negatives) - кількість правильно передбачених негативних класів (об'єкти, які не відслідковуються, і модель правильно це передбачила),  
 FP (False Positives) - кількість неправильно передбачених позитивних класів (об'єкти невірно відслідковані),  
 FN (False Negatives) - кількість неправильно передбачених негативних класів (об'єкти, які відслідковуються, але модель помилилася).

$$\text{Втрати} = \frac{1}{n} \sum_{i=1}^n (\text{predictions}_i - \text{actual}_i)^2, \quad (2)$$

- де  $\text{predictions}_i$  - передбачене значення моделі для  $i$ -го прикладу,  
 $\text{actual}_i$  - справжнє значення для  $i$ -го прикладу, яке модель мала передбачити,  
 $n$  - кількість значень у вибірці.

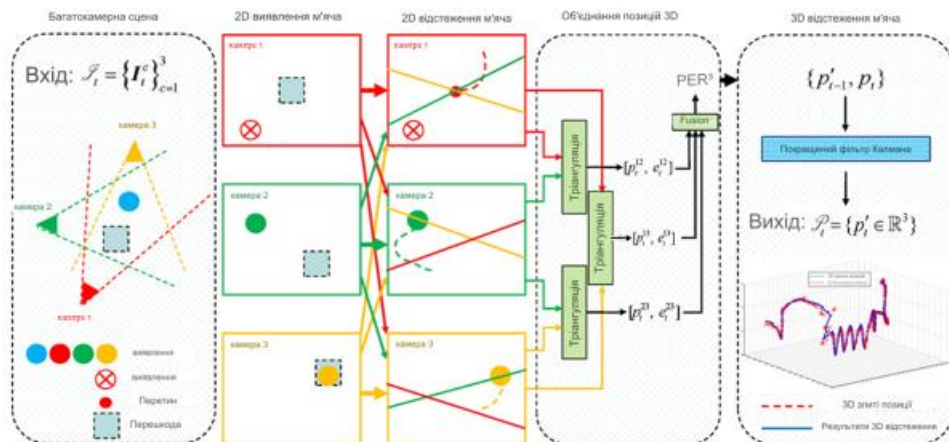
4

## СХЕМА МЕТОДУ ГЛИБОКОГО НАВЧАННЯ НЕЙРОМЕРЕЖ



5

## СХЕМА МЕТОДУ БАГАТОКАМЕРНОГО ТРЕКІНГУ ОБ'ЄКТІВ



6

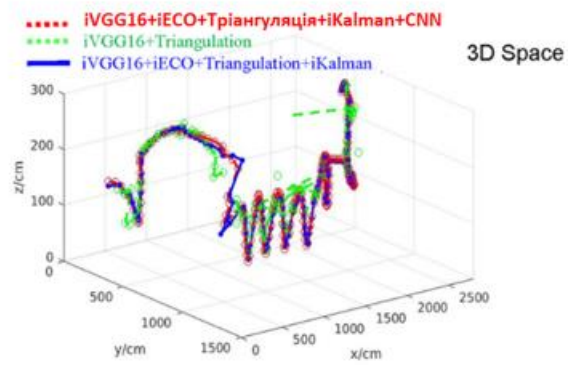


## Демонстрація роботи методу на реальних прикладах



7

## РЕЗУЛЬТАТ МОДЕЛЮВАННЯ МЕТОДІВ В 3D ПРОСТОРІ



8

**РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ ЗАСТОСУВАННЯ ГЛИБОКОГО  
НАВЧАННЯ ДО МЕТОДІВ БАГАТОКАМЕРНОГО  
ВІДСЛІДКУВАННЯ ОБ'ЄКТІВ**

Метод	Точність
RANSAC	50 %
MIP-з-обмеженнями	62 %
Система вимірювання положення м'яча	58 %
VGG16+Триангуляція	56 %
iVGG16+Триангуляція (3D)	69 %
iVGG16+iKCF+Триангуляція	60 %
iVGG16+iECO+Триангуляція (3D)	79 %
iVGG16+iECO+Триангуляція+Калмана	81 %
iVGG16+iECO+Триангуляція+iKalman (3D)	92 %
<b>iVGG16+iECO+Триангуляція+iKalman+CNN</b>	<b>95 %</b>

9

**ВИСНОВКИ**

1. Проаналізовано існуючі підходи до багатокамерного відслідковування об'єктів. Виявлено, що за комбінування існуючих підходів виходить покращити точність відслідковування об'єктів, завдяки вдалому використанню сильних сторін різних методів.
2. Проаналізовано приклади застосування методів багатокамерного трекінгу разом з методами машинного навчання. Такі комбінації показують свою широку ефективність в різних сферах: безпеки, бізнесу, медицини, спорту тощо.
3. Вибрано приклади з гарною потенційною сумісністю з технологією глибокого навчання. А саме застосування глибокого навчання для рішень задач відслідковування м'яча в тривимірному просторі виправдало очікування і показало кращі результати точності 95%.
4. Вдосконалено підхід до багатокамерного відслідковування об'єктів завдяки використанню методу машинного навчання. Доповнення комбінації методів технологією згорткової нейронної мережі є вдалим модифікуванням попередньої моделі з точністю 92%.

10

## ПУБЛІКАЦІ ТА АПРОБАЦІЯ РОБОТИ

### Статті:

1. Буряк Р.В., Золотухіна О.А. Питання розробки методів детектування об'єктів при багатокамерному трекінгу на основі глибокого навчання// Зв'язок. Подано до друку.

### Тези доповідей:

1. Буряк Р.В., Золотухіна О.А. Використання машинного навчання і технології багатокамерного трекінгу для вирішення побутових і промислових задач.// I Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу». – Київ: ДУІКТ, 2023. – С. 211.

2. Буряк Р.В., Золотухіна О.А. Особливості розробки технологій машинного навчання та відстеження об'єктів для цифровізації життя у місті. // I Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу». – Київ: ДУІКТ, 2023. – С. 212.

11

**ДЯКУЮ ЗА УВАГУ!**