

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

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Розробка методу відстеження траєкторії людини в натовпі на основі технологій комп'ютерного зору»

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*

\_\_\_\_\_ Ангеліна УСІЧЕНКО  
(підпис)

Виконала: здобувачка вищої освіти групи ПДМ-62

\_\_\_\_\_ Ангеліна УСІЧЕНКО

Керівник: \_\_\_\_\_ Олена НЕГОДЕНКО  
к.т.н., доцент

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

**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**

\_\_\_\_\_ Усіченко Ангеліні Андріївні

1. Тема кваліфікаційної роботи: «Розробка методу відстеження траєкторії людини в натовпі на основі технологій комп'ютерного зору»

керівник кваліфікаційної роботи Олена НЕГОДЕНКО к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023 р. №145.

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

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, параметри комп'ютерного зору, існуючі методи відстеження людини.

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

1. Дослідження принципів методу відстеження людини.

2. Розробка вимог до методу відстеження траєкторії людини в натовпі.

5. Перелік графічного матеріалу: *презентація*

1. Загальна характеристика методів комп'ютерного зору.
2. Відстеження траєкторії на основі мануального налаштування.
3. Порівняння засобів візуального відстеження об'єктів.
4. Виявлення осіб на зображенні.
5. Ідентифікація особи.

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 алгоритмів та технологій, що використовуються для відстеження траєкторії людини	06.11-12.11.23	
3	Дослідження наявних підходів	13.11-19.11.23	
4	Побудова досліджуваної математичної моделі	20.11-26.11.23	
5	Розробка методу відстеження траєкторії людини в натовпі	27.11-03.12.23	
6	Обробка результатів роботи алгоритму відстеження	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	

Здобувачка вищої освіти

\_\_\_\_\_ (підпис)

Ангеліна УСІЧЕНКО

Керівник кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Олена НЕГОДЕНКО





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 70 с., 3 табл., 22 рис., 30 джерел.

*Мета роботи* – підвищення ефективності відстеження та формування відповідних траєкторій переміщення осіб за допомогою технологій комп'ютерного зору.

*Об'єкт дослідження* – відстеження людей у натовпі на основі відеоматеріалів.

*Предмет дослідження* – технології комп'ютерного зору для фіксації траєкторії відстежуваних осіб.

*Методи дослідження* – аналіз стабільності ідентифікаторів об'єктів; машинне навчання, інтерпретація геопросторових даних.

*Короткий зміст роботи:* Проведено аналіз побудови і функціонування згорткових нейронних мереж, їх використання для виявлення людей на зображенні та можливі варіації, придатність яких порівнюється в контексті обробки відеозапису, знятого з дрона, тобто при перегляді з висоти і з різних ракурсів.

Синтезовано методи визначення GPS-координат відстежуваних осіб без використання спеціального обладнання для мобільної камери. Також описано загальні принципи методів повторної ідентифікації осіб і конкретні шляхи реалізації відповідних задач.

У ході проведених досліджень випробувано 4 різних варіанти детекторів, та проведено аналіз ефективності їх застосування для різних ситуацій. Для отриманих експериментів було обрано найбільш раціональний варіант, який на тестовому наборі даних досягає середньої точності.

На основі результатів виконаних досліджень сформовано набір ідентифікаторів, що містить 384 зображення та 5 відповідних осіб. Де обробка даних виконувалась не лише базовими методами повторної ідентифікації за допомогою гістограм, а й за допомогою машинного навчання, а саме нейронної мережі *AlignedReID*, яка дає показник успішності 95,2% на тестовому наборі даних навіть без використання додаткових коригувань зображення. І навпаки, підхід, заснований на гістограмі, досяг максимуму приблизно в 60% навіть з коригуванням.

**КЛЮЧОВІ СЛОВА:** КОМП'ЮТЕРНИЙ ЗІР, АВТОМАТИЧНА ОБРОБКА ЗОБРАЖЕНЬ, ВИЯВЛЕННЯ ЛЮДЕЙ У НАТОВПІ, ВІДСТЕЖЕННЯ ТРАЄКТОРІЇ ОБ'ЄКТІВ, МАШИННЕ НАВЧАННЯ, ДРОН, GPS, GOOGLE EARTH.

## **ABSTRACT**

Text part of the master's qualification work: 70 pages, 22 pictures, 3 table, 30 sources.

The purpose of the work is a development of a method of tracking a person's trajectory using computer vision technologies.

Object of research – methods of tracking people in a crowd based on video footage.

Subject of research – application of computer vision technologies to fix the trajectory of tracked persons.

Summary of the work: An analysis of the construction and functioning of convolutional neural networks, their use for detecting people in the image and possible variations is carried out, the suitability of which is compared in the context of processing video footage taken from a drone, that is, when viewed from a height and from different angles.

Methods for determining the GPS coordinates of tracked persons without the use of special equipment for a mobile camera are synthesized. The general principles of methods of re-identification of persons and specific ways of realizing the relevant tasks are also described.

In the course of the research, 4 different variants of detectors were tested, and the effectiveness of their application for different situations was analyzed. For the experiments, the most rational variant was chosen, which achieves average accuracy on the test data set.

Based on the results of the research, a set of identifiers was formed, containing 384 images and 5 corresponding faces. Where data processing was performed not only by basic methods of re-identification using histograms, but also by machine learning, namely the AlignedReID neural network, which gives a success rate of 95.2% on the test dataset even without using additional image adjustments. Conversely, the histogram-based approach reached a maximum of about 60% even with the adjustment.

**KEYWORDS:** COMPUTER VISION, AUTOMATIC IMAGE PROCESSING, DETECTION OF PEOPLE IN A CROWD, TRACKING OF OBJECTS TRAJECTORY, MACHINE LEARNING, DRONE, GPS, GOOGLE EARTH.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	12
1.1 Загальна характеристика методів комп'ютерного зору для відстеження об'єктів.....	12
1.2 Опис алгоритмів та технологій, що використовуються для відстеження траєкторії людини .....	15
1.2.1 Класифікація засобів відстеження траєкторії об'єкта.....	15
1.2.2 Відстеження траєкторії на основі мануального налаштування .....	19
1.2.3 Відстеження траєкторії людини засобами глибокої обробки .....	21
1.3 Переваги та недоліки різних підходів.....	28
1.3.1 Порівняльні набори даних .....	28
1.3.2 Порівняння засобів візуального відстеження об'єктів .....	30
РОЗДІЛ 2 РОЗРОБКА МЕТОДУ ВІДСТЕЖЕННЯ ТРАЄКТОРІЇ ЛЮДИНИ В НАТОВПІ НА ОСНОВІ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ .....	33
2.1 Побудова досліджуваної математичної моделі .....	33
2.2 Розробка методу відстеження траєкторії людини в натовпі за допомогою технологій комп'ютерного зору .....	41
РОЗДІЛ 3 ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ РОЗРОБЛЮВАНОВОГО МЕТОДУ.....	59
3.1 Опис експериментів для перевірки працездатності та ефективності методу	59
3.2 Особливості проведення ексериментів.....	62
3.3 Аналіз результатів обробки даних .....	74
ВИСНОВКИ.....	78
ПЕРЕЛІК ПОСИЛАНЬ .....	80
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	84



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

VBO	–	візуальне відстеження об'єктів
ГНМ	–	глибока нейронна мережа
ЗНМ	–	згорткова нейронна мережа
РНМ	–	рекурентна нейронна мережа
<i>GPS</i>	–	<i>global positioning system</i>
<i>HOG</i>	–	<i>histogram of oriented gradients</i>
<i>ACF</i>	–	<i>aggregate channel functions</i>
<i>EEN</i>	–	<i>end-to-end network</i>
<i>FEN</i>	–	<i>feature extraction network</i>

## ВСТУП

**Актуальність теми.** Використання сучасних технологій дає змогу досліджувати і розвивати різні сфери суспільства. Тут, у першу чергу, можна відзначити питання безпеки, а саме моніторингу переміщення певних осіб та контроль статусу визначених територій. Наразі для цього все частіше використовуються безпілотні літальні апарати (дрони), які можуть, наприклад, контролювати або обстежувати території на безпечній відстані та без безпосередньої участі спостерігача [1]. Таким чином, дрони виконують фіксацію відеоматеріалів, які, у більшості випадків, все ще потребують подальшої ручної обробки відповідними фахівцями. Так, актуальним є питання дослідження методів комп'ютерного зору для автоматичної обробки великих об'ємів даних з можливістю ефективного відстеження осіб та формування відповідних траєкторій переміщення.

**Мета роботи і задачі дослідження.** Підвищення ефективності відстеження та формування відповідних траєкторій переміщення осіб за допомогою технологій комп'ютерного зору. Досягнення вказаної мети забезпечується розв'язуванням наступних задач:

1. Провести огляд методів комп'ютерного зору для відстеження об'єктів.
2. Описати алгоритми та технології, що використовуються для відстеження траєкторії людини.
3. Визначити основні переваги та недоліки аналізованих методів.
4. Сформуванати математичну модель обробки даних.
5. Визначити основні інструменти та засоби обробки інформації для візуалізації траєкторії відстежуваних осіб.
6. Удосконалити метод обробки даних.
7. Провести опис експериментів для перевірки працездатності та ефективності розроблюваного методу.
8. Проаналізувати результати обробки даних.
9. Скласти висновки за результатами дослідження.

Ця магістерська робота присвячена питанню відстеження людей на відео, знятому дроном. Використовуючи комп'ютерний зір, він намагається автоматично виявити пішоходів на зображенні. На основі інформації, отриманої з дрона, і запису він потім обчислює положення людей у реальному світі у формі GPS-координат.

**Об'єкт дослідження** – відстеження траєкторії людей у натовпі на основі відеоматеріалів.

**Предмет дослідження** – технологія комп'ютерного зору для фіксації траєкторії відстежуваних осіб.

**Методи дослідження:** аналіз стабільності ідентифікаторів об'єктів; машинне навчання, інтерпритація геопросторових даних.

Розроблюваний метод передбачає, що для виявлених осіб на зображенні створюються відповідні маркери. Нові виявлення в інших зображеннях додаються цим особам. Потім вектори руху зберігається як траєкторія *GPS* для відстеження відповідного переміщення на картах геоінформаційних систем.

Таким чином, наведений алгоритм обробки відеоматеріалів може значно полегшити вже згадані зусилля щодо моніторингу територій та переміщення людей на них, що дає можливість підвищити безпеку в різних сферах комерційної та державної діяльності [2].

**Наукова новизна одержаних результатів.** У поточній роботі розроблено метод та відповідний математичний апарат не лише виявлення осіб на основі відеоматеріалів, а і візуалізації відповідних траєкторій засобами геоінформаційних систем. Також, тут передбачено застосування опцій для корегування отриманих траєкторій відповідно до використовуваних геопросторових баз даних.

**Практичне значення одержаних результатів.** З урахуванням сучасних реалій, результати даного дослідження дозволяють розширити практичну базу для розробки комплексних автоматизованих рішень щодо задач моніторингу державного кордону та окремих територій з наявними об'єктами критичної інфраструктури.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Загальна характеристика методів комп'ютерного зору для відстеження об'єктів

Візуальне відстеження об'єктів (ВВО) - це процес безперервної оцінки стану цілі у відеопослідовності від її першої появи до останньої. Він привернув велику увагу завдяки широкому використанню в багатьох додатках. ВВО є добре відомою дослідницькою проблемою комп'ютерного зору, яка проводиться протягом кількох десятиліть. Однак, у точному відстеженні людей є ряд ускладнень через зміни зовнішнього вигляду, варіації пози, оклюзії, варіації освітлення, розмиття руху та шуму на фоні (рис. 1.1).

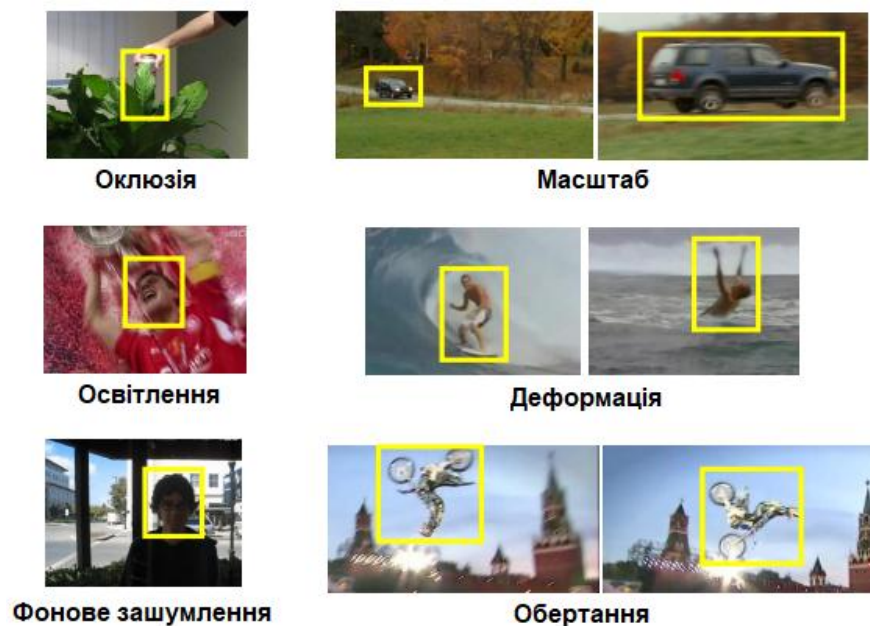


Рис. 1.1. Проблеми візуального відстеження об'єктів

Фреймворки ВВО можна класифікувати на основі представлення моделі як трекери точок, трекери зовнішнього вигляду та трекери силуетів. Під час відстеження точок цільові об'єкти представлені точками в послідовних кадрах, а

зв'язок точок використовується для прогнозування руху та розташування цілі. Трекери точок [3] зазвичай використовуються, коли цільові об'єкти займають дуже малі області у відеопослідовності, і їх розмір не є важливим фактором. У силуетних трекарах цілі представлені регіонами карти країв або контурами периметру. Область всередині периметру називається силуетом, і така схема використовується для зображення складних нежорстких форм.

Трекери на основі зовнішнього вигляду використовують візуальні функції для представлення цільових об'єктів. Модель зовнішнього вигляду цільового об'єкта та попередній стан цієї цілі використовуються для прогнозування цільового стану в поточному кадрі. Оскільки розмір і орієнтація цілі також можуть бути передбачені трекарами зовнішнього вигляду, кілька чутливих до безпеки програм зацікавлені в цих підходах, таких як відеоспостереження, виявлення аномальних подій, моніторинг трафіку, а також інші області, такі як медична візуалізація [4].

Візуальні засоби відстеження об'єктів також можна класифікувати на онлайн та автономні (офлайн) на основі режиму обробки кадрів. У онлайн-відстеженні структура відстеження передбачає цільовий стан у поточному кадрі, обробляючи минулу інформацію, доступну до поточного кадру. Навпаки, деякі трекари отримують інформацію з минулих і майбутніх кадрів, щоб передбачити цільовий стан. Зазвичай, у той час як онлайн-трекари передбачають цільовий стан з меншою пам'яттю та обчислювальними витратами, одним трекарам потрібна величезна кількість обчислювальних ресурсів, оскільки вони обробляють усі кадри одночасно. Офлайн трекари не підходять для завдань онлайн, оскільки вони не дають результатів режимі реального часу.

Трекари зовнішнього вигляду використовують створені вручну (мануальні) або «глибокі» риси для представлення цільової моделі. Створені вручну трекари на основі функцій витягують інформацію із зображень відповідно до певного заздалегідь заданого вручну алгоритму на основі експертних знань. Гістограма орієнтованого градієнта (англ. *HOG*), характеристика сукупного каналу (англ. *ACF*), локальний бінарний шаблон (англ. *LBP*) і масштабне інваріантне перетворення ознак (англ. *SIFT*) є одними з прикладів для елементів мануальної

обробки, які використовуються в ВВО. Незважаючи на те, що поява глибокого навчання продемонструвала значні покращення в багатьох областях застосування, створені вручну функції все ще використовуються у візуальному відстеженні з двох основних причин. По-перше, глибоке навчання вимагає величезної кількості даних для навчання моделі, і це часто недоступно у ВВО. По-друге, через високу обчислювальну вартість глибокого навчання відстеження в реальному часі нелегко досягти. Таким чином, мануальне відстеження все ще потребує досліджень, щоб створити надійні системи відстеження об'єктів у реальному часі.

Завдяки швидкому збільшенню обчислювальної потужності глибокі нейронні мережі (ГНН) можуть напряду вивчати функції з необроблених даних, не вдаючись до ручної обробки. ГНН, особливо згорткові нейронні мережі (ЗНМ), продемонстрували сучасну продуктивність у кількох задачах комп'ютерного зору. Однак небагато фреймворків візуального відстеження використовують ЗНМ. Основна причина цього полягає в тому, що ЗНМ використовують широкий набір параметрів, вони вимагають величезних обсягів навчальних даних, які стали доступними для ВВО відносно недавно.

Більшість трекерів зовнішнього вигляду на основі глибоких функцій [6] керують дефіцитом навчальних даних, передаючи одні вивчені функції ГНН до онлайн-відстеження. Ці підходи отримують загальні характеристики зображення із завдань класифікації об'єктів. Однак вони не вивчають подібну локальну структуру та інформацію про внутрішню геометричну структуру цілей. Кілька підходів до глибокого відстеження [6] аналізують внутрішні властивості функцій ЗНМ з точки зору візуального відстеження та пропонують відповідні алгоритми відстеження. Незважаючи на те, що ці підходи продемонстрували найсучаснішу продуктивність, вони характеризуються певною надмірністю, оскільки вони налаштовані онлайн з меншою кількістю зразків. Інша група глибоких трекерів на основі зовнішнього вигляду збільшує швидкість відстеження за допомогою фіксованих ЗНМ. Вони тренують свою модель і уникають онлайн-адаптації, щоб збільшити швидкість відстеження до реального часу. Вони або використовують підхід навчання подібності, або мережу регресії, щоб передбачити місцезнаходження цілі. Однак

їхня точність значно нижча, оскільки вони уникають важливих елементів на відео під час відстеження.

Надійний засіб відстеження об'єктів повинен мати найсучаснішу точність, зберігаючи швидкість відстеження в реальному часі, оскільки це вимоги для багатьох реальних програм. Незважаючи на те, що мануальні трекери є економічно ефективними і, отже, можуть відстежувати в режимі реального часу, їх точність відстеження нижча, оскільки вони не можуть захопити семантичну інформацію цілей і не стійкі до значних змін зовнішнього вигляду. З іншого боку, глибокі трекери демонструють найсучаснішу точність завдяки своїм багатим можливостям представлення ієрархічних функцій рівня. Однак швидкість їх відстеження дуже низька через обчислювальну вартість оновлення великої кількості параметрів ЗНМ.

## **1.2 Опис алгоритмів та технологій, що використовуються для відстеження траєкторії людини**

Візуальне відстеження траєкторії людини викликає великий інтерес у багатьох сферах застосування, таких як відеоспостереження, робототехніка, аналіз спортивного відео, доповнена реальність, розпізнавання активності, автономне водіння, моніторинг трафіку, взаємодія людини з комп'ютером та медичні діагностичні системи. Надійна відповідна структура зможе відстежувати конкретну ціль у багатьох викликах, таких як різні пози, оклюзії, зміни освітлення, варіації масштабу, низька роздільна здатність, розмиття руху, варіації точки зору та фону. Незважаючи на те, що дослідження відстеження об'єктів проводяться протягом кількох десятиліть, ВВО залишається активною темою дослідження через високі вимоги та виклики [7].

### **1.2.1 Класифікація засобів відстеження траєкторії об'єкта**

Візуальні засоби відстеження траєкторії об'єктів використовують візуальні функції для моделювання цільових об'єктів. Типова структура відстеження має п'ять основних компонентів, таких як ініціалізація цілі, моделювання зовнішнього

вигляду, пошук моделі, локалізація цілі та оновлення моделі. Початковий стан (розташування та розмір) цілі зазвичай вказується в першому кадрі відеопослідовності, а прямокутні рамки використовуються для повторного відображення стану цілі. Компонент моделювання зовнішнього вигляду складається з двох підкомпонентів: візуального представлення та статистичного вимірювання. Візуальне представлення виділяє візуальні особливості цілі, мануальні чи глибокі. З іншого боку, компонент статистичних вимірювань використовується для обчислення подібності між різними спостереженнями. Потім компонент пошуку моделі шукає цільову модель у наступних кадрах і виводить оцінку відповіді. Нарешті, на основі оцінки визначається стан цілі в поточному кадрі. Компонент оновлення моделі використовується для оновлення моделі зовнішнього вигляду новим спостереженням або зразком цілі.

Перегляд і оцінка візуальних засобів відстеження є дуже важливими для виявлення прогалин у дослідженні та пошуку майбутніх напрямків. Існує кілька досліджень [8], які були проведені для класифікації цих засобів на основі різних категорій та аналізу їх продуктивності.

Автори [9] розглянули візуальні засоби, класифікувавши їх на три основні групи: трекари точок, трекари силуетів і трекари зовнішнього вигляду. У роботі [10] проведено експериментальне дослідження дев'ятнадцяти трекарів зовнішнього вигляду, оцінивши їх продуктивність на наборі даних *ALOV300++*. Проведено оцінку продуктивності 31 сучасного трекара на наборі даних *Object Tracking Benchmark (OTB)* [11].

В основу класифікації досліджуваних засобів покладено відстеження одного та кількох об'єктів. У той час як одна ціль обробляється при відстеженні одного об'єкта, кілька зразків певного класу об'єктів відстежуються при відстеженні кількох об'єктів. Фреймворки відстеження окремих об'єктів зосереджені головним чином на розробці складних моделей зовнішнього вигляду для вирішення таких проблем, як варіації масштабу, зміни освітлення, низька роздільна здатність і фоновий шум. На додаток до цих загальних проблем, багатооб'єктні системи відстеження стикаються з іншими проблемами, такими як асоціації між кількома



зразками, важкі оклюзії, ініціалізація та завершення цілей і подібні відволікаючі фактори. Як правило, засоби відстеження одного об'єкта в основному залежать від моделі зовнішнього вигляду цілі [12]. Навпаки, багатооб'єктні засоби розглядають модель зовнішнього вигляду як одну з ознак серед кількох інших, таких як модель руху та модель асоціації.

У той час як багатооб'єктні трекери в основному пропонуються для відстеження пішоходів [13], однооб'єктні трекери пропонуються для всіх класів об'єктів, таких як люди, транспортні засоби, літаки, тварини та ін. Таким чином, розробка системи відстеження траєкторії кількох об'єктів є складнішою, ніж структура відстеження одного об'єкта, оскільки вона стикається з кількома складними проблемами.

Засоби відстеження також можна класифікувати як модельні та безмодельні на основі цільової ініціалізації. При відстеженні на основі моделі, попередньо навчений детектор об'єктів, що спеціалізується на класі, використовується для ініціалізації цілі в її початковому кадрі, а потім відстеження здійснюється в інших кадрах на основі цієї ініціалізації. Навпаки, безмодельні засоби відстеження об'єктів, вимагають ручної ініціалізації на початковому кадрі та відповідного визначення цілі в решті кадрів. Продуктивність відстеження на основі моделі в основному залежить від продуктивності попередньо навченого детектора об'єктів.

Офлайн та онлайн-відстеження є ще одним набором класифікацій у техніці, заснованій на техніці комп'ютерного зору. Схема роботи засобів відстеження у цих режимах наведена на рис. 1.2.

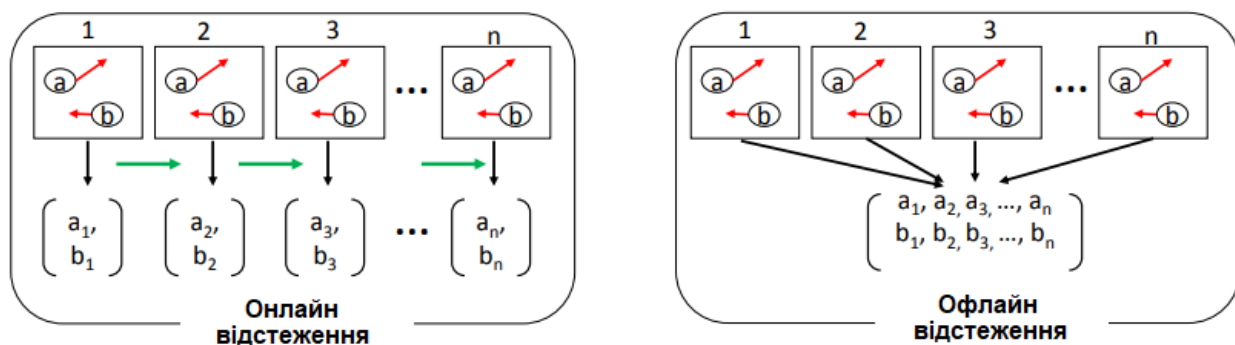


Рис. 1.2. Схематичне пояснення відстеження онлайн та офлайн

У онлайн-відстеженні стан цілі в поточному кадрі прогнозується шляхом обробки попередньої інформації, доступної поточному кадру. Окремі засоби використовують минулі, поточні та майбутні спостереження, щоб передбачити стан цілі в поточному кадрі.

Як показано на рис. 1.2 (ліворуч), онлайн-трекери обробляють кадри послідовно. Онлайн-трекери передбачають стан цілі в поточному кадрі, обробляючи минулу інформацію, доступну поточному кадру. Навпаки, як показано на рис. 1.2 (праворуч), окремі трекери використовують минулі, поточні та майбутні спостереження, щоб передбачити стан цілі в поточному кадрі. Відстеження у режимі реального часу не підходить для багатьох програм, оскільки вони затримують виведення прогнозів.

Більшість трекерів можна класифікувати як генеративні та дискримінаційні на основі техніки представлення моделі. Генеративні трекери вивчають зовнішній вигляд цілі та зосереджуються на тому, як точно використовувати вивчену модель у майбутніх кадрах. Ці методи здатні відстежувати цільовий об'єкт шляхом пошуку найкращого кандидата, який найбільше схожий на еталонний шаблон у кожному кадрі. Генеративні моделі зосереджені на навчанні візуального представлення цільової області об'єкта, ігноруючи фон. Як правило, ці моделі забезпечують надійну роботу в менш складних середовищах завдяки своїй техніці представлення. З іншого боку, на них часто впливають дистрактори, які мають зовнішній вигляд, подібний до цільового класу об'єктів, створених фоновими областями. Навпаки, дискримінаційні трекери відстежують ціль, вивчаючи модель зовнішнього вигляду яка має на меті дискримінаційне збільшення поділу між цільовими та нецільовими регіонами. Більшість дискримінаційних трекерів трактують відстеження як проблему бінарної класифікації, а онлайн-навчений класифікатор використовується для розрізнення цілі від фону. Ці підходи більш надійні, ніж генеративні трекери в сценах оклюзії, оскільки вони включають фонові області в процес навчання. Однак перекриття є загальною проблемою дискримінаційного відстеження [14].

Як зазначалося раніше, сучасні засоби відстеження також можна класифікувати як мануальні та глибокі на основі їх візуального представлення функцій. Під час мануального відстеження, інформація про зовнішній вигляд цільового об'єкта витягується відповідно до заздалегідь заданого вручну алгоритму на основі експертних знань. Навпаки, фреймворки глибокого відстеження безпосередньо вивчають функції з необроблених даних, не вдаючись до ручного створення. Мануальні елементи здатні фіксувати лише певний шаблон візуальної інформації, тому вони не є узагальненими. Вони не в змозі вловити семантичну інформацію цілей і не стійкі до значних змін зовнішнього вигляду. Тому їх точність відстеження нижча порівняно з глибоким відстеженням [15]. Навпаки, оскільки глибокі нейронні мережі використовують широкий набір параметрів, вони вимагають величезних обсягів навчальних даних, а їх обчислювальна вартість висока.

### **1.2.2 Відстеження траєкторії на основі мануального налаштування**

Цільове представлення є одним із основних компонентів усіх фреймворків відстеження на основі зовнішнього вигляду. Оскільки методи візуального відстеження досліджували протягом кількох десятиліть, було запропоновано численні схеми представлення [11]. До появи глибокого відстеження всі трекери зовнішнього вигляду поклалися на створені вручну функції для представлення цільового об'єкта. Трекери, створені вручну, використовують експерт домену щоб витягти ознаки з необроблених даних, а потім перенести витягнуті ознаки в орієнтовану систему, часто класифікатор, для відстеження цілі. Мануальні трекери також називають традиційним трекінгом [10; 11].

Розріджене представлення [16] є однією зі схем представлень у мануальному відстеженні. У цій техніці цільовий кандидат представлений як розріджена лінійна комбінація шаблонів, які можна динамічно оновлювати для підтримки актуальної моделі зовнішнього вигляду цільової програми. Показано, що ця схема представлення є надійною проти часткових оклюзій, що призводить до покращеної продуктивності відстеження.

Деякі трекеери використовують кольорову гістограму для представлення моделі зовнішнього вигляду цілі. Автори [17] запропонували алгоритм відстеження на основі кольорової гістограми з алгоритмом середнього зсуву. У цьому трекеері відповідний коефіцієнт використовується для вимірювання подібності між потенційними кандидатами, а алгоритм середнього зсуву використовується для ефективного пошуку між цілями. Оскільки трекеери колірної гістограми ігнорують форму та текстуру цілей, їх продуктивність обмежена.

Гістограма орієнтованих градієнтів (*HOG*) використовувалась у багатьох трекеерах через їх хорошу здатність відображати спрямовані локальні краї. Також використовували локальні двійкові шаблони (англ. *LBP*) для цільового представлення [18].

Кілька засобів відстеження об'єктів [13] використовували функції сукупного каналу (*ACF*) через їх сучасну продуктивність у задачах виявлення об'єктів і високу швидкість роботи. У поданні *ACF* візуальні шаблони цілі обчислюються як десять різних каналів, таких як нормалізована величина градієнта, *HOG* (шість каналів) і кольори *LUV* (три канали).

Автори [13] запропонували багатооб'єктну структуру відстеження з класифікатором *ACF* і *AdaBoost*. Вони також запропонували техніку онлайн-навчання детектора індивідуальних категорій, яка називається *IDBoost*. У цій техніці попередньо навчений підвищений детектор категорії використовується для ініціалізації окремого детектора в режимі онлайн з найменшими обчислювальними витратами. В *IDBoost* індивідуальний детектор ініціалізується за допомогою того самого каскаду функцій, що й детектор категорії, разом із елементарними маніпуляціями з пороговими значеннями слабких класифікаторів. Ця техніка оцінюється на базі даних розпізнавання облич і пішоходів та показала швидше навчання та кращу точність розпізнавання.

У роботі [19] запропоновано генеративний трекеер, який вивчає низьковимірне підпросторове представлення цілі. Їхній підхід ефективно оновлює цільову модель онлайн за допомогою інкрементального *PCA*.

Більшість мануальних засобів відстеження [13; 20] використовують техніку відстеження шляхом виявлення та показали найсучасніші характеристики. Метод відстеження за виявленням розглядає відстеження об'єкта як безперервний процес виявлення. У більшості ситуацій бінарний класифікатор навчається в режимі онлайн шляхом захоплення позитивних і негативних зразків із цільової та фонові областей. Максимальний відгук класифікатора використовується для визначення місцезнаходження цілі в новому кадрі. Потім класифікатор оновлюється новими зразками, щоб зафіксувати зміни зовнішнього вигляду цілі. Кілька добре відомих класифікаторів використовуються з цією технікою [20].

У підсумку, численні схеми представлення ознак пропонують мануальне відстеження. Тут продуктивність відповідних засобів в основному залежать від методів виділення ознак. Серед кількох представлень функцій, *ACF* можна використовувати для візуального відстеження людини завдяки можливості навчання та високій швидкості виділення функцій. Оскільки мануальні функції розроблені на основі певного алгоритму або експертних знань, вони не можуть добре узагальнюватися, а тому їх продуктивність обмежена. Однак майже всі мануальні засоби відстеження [11] здатні працювати з меншими обчислювальними витратами, і тому вони все ще перебувають у сфері активних досліджень.

### **1.2.3 Відстеження траєкторії людини засобами глибокої обробки**

Більшість традиційних трекерів використовують мануальні функції для моделювання траєкторії цілі. Оскільки мануально налаштовані об'єкти не можуть відобразити семантичну інформацію цілей, вони не є стійкими, щоб зафіксувати значні зміни зовнішнього вигляду цілей. Продуктивність мануальних засобів відстеження далека від вимог реальних програм. Таким чином, потрібні складні методи навчання, щоб покращити представлення функцій структур відстеження.

Побудовані за принципом системи біологічного навчання, глибокі нейронні мережі [7; 8] привернули велику увагу в машинному навчанні, оскільки вони створюють найсучасніші рішення для різних застосувань. Тут ГНН вказує на багаторівневу нейронну мережеву архітектуру, яка може ефективно вивчати

функції з необроблених даних. Продуктивність мануальних трекерів: *Struck*, *CSK*, *KMS*, *SBT* і *CT* порівнюється з глибокими трекерами: *MDNe*, *SA SIAM*, *DSiamM* і *SINT* за тестом *OTB2018* [21].

Мотивовані великим успіхом ГНН в інших завданнях, дослідники починають розробка структур відстеження з ГНН, які називають глибоким відстеженням. Глибинні трекери показали значне покращення продуктивності, ніж мануальні трекери. Порівняння продуктивності на *OTB2018* [21] чітко показує, що глибокі трекери значно перевершують мануальні з великим запасом. Однак глибокі трекери все ще мають бути доопрацьованими для досягнення швидкості відстеження в реальному часі, оскільки вони обробляють величезну кількість папараметрів.

Згорткова нейронна мережа (ЗНМ) і мережа довгострокової пам'яті (англ. *LSTM*) є двома основними архітектурами ГНН, які навчені до роботи з багатовимірними та послідовними даними відповідно. І ЗНМ, і *LSTM* використовуються у ВВО. Однак зовнішній вигляд базується на фреймворках глибокого відстеження в основному залежать від архітектури ЗНМ через їх хорошу продуктивність.

ЗНМ демонструє найсучаснішу продуктивність у кількох завданнях завдяки багатій можливості ієрархічного представлення функцій рівня. За останні роки запропоновано кілька архітектур ЗНМ, таких як *AlexNet*, *VGG-Net*, *GoogLeNet* і *ResNet* [22]. Типова архітектура згорткової мережі складається зі згорткових рівнів і повністю зв'язаних рівнів. В архітектурі ЗНМ ознаки виділяються через ієрархічні згорткові шари, а повністю зв'язані рівні використовуються для обчислення балів класифікації. ЗНМ визначаються шляхом згортання ряду суміжних одиниць введення з функцією активації, вихід якої визначає наступний рівень у мережі, і виконання цього для всіх суміжних одиниць.

Навчальна здатність ЗНМ в основному залежить від кількості навчальних даних. Більша кількість навчальних вибірок збільшує здатність мережі до узагальнення, оскільки ЗНМ мають справу з величезною кількістю параметрів. Навчання архітектури ЗНМ з меншою кількістю зразків призведе до

перенасичення. Таким чином, нестача даних є основною проблемою глибокого відстеження, оскільки ціль невідома, а доступних зразків дуже мало.

Можливість узагальнення архітектури ЗНМ буде обмеженою, якщо її навчати з меншою кількістю зразків. Методи глибокої адаптації домену [22] вирішують проблему нестачі даних шляхом передачі отриманих знань добре навченої (вихідної) мережі до іншої пов'язаної (цільової) мережі з меншою кількістю позначених зразків. Ця техніка мінімізує відстань між вихідним і цільовим розподілами. У адаптації джерела до цільової мережі ранні згорткові шари вихідної мережі заморожуються та передаються для представлення цільової мережі, оскільки їхні функції є більш загальними. Оскільки глибші згорткові функції є менш піддатливими для передачі, вони вивчаються шляхом повторного налаштування для представлення цільової мережі. У їхній техніці повністю підключені рівні цільової мережі вивчаються за допомогою багатоядерної техніки максимальної середньої розбіжності (англ. *MMD*). Автори [22] запропонували подібну техніку, яка використовує втрату зв'язків домену для навчання цільової мережі з вихідної мережі. Ці методи адаптації глибокої області можуть бути використані для глибокого відстеження, оскільки брак даних значно обмежує продуктивність.

Архітектури глибокого відстеження можна класифікувати та переглядати на основі використання ними ЗНМ. Глибоке відстеження можна класифікувати як відстеження мережею вилучення функцій (англ. *FEN*) і відстеження наскрізною мережею (англ. *EEN*). У відстеженні *FEN*, ЗНМ використовуються лише для виділення глибинних особливостей, а потім для визначення місцезнаходження використовується традиційний метод навчання. Трекери *FEN* використовують попередньо навчені архітектури ЗНМ, які спочатку навчені для завдань класифікації об'єктів. Навпаки, трекери *EEN* розробляють і навчають архітектури ЗНМ спеціально для ВВО. Більшість трекерів *EEN* дотримуються двох різних стратегій. У той час як перша група трекерів тренується та використовує дискримінаційну класифікаторну мережу, інша група трекерів розглядає відстеження як проблему зіставлення подібності.

Відстеження траєкторії об'єкта є складним завданням, оскільки трекеру надається лише один зразок цілі, і він повинен передбачити варіації зовнішнього вигляду цілі в наступних кадрах. Тому надійний трекер повинен правильно моделювати зовнішній вигляд цілі за допомогою доступних зразків.

Щоб впоратися з нестачею даних, група трекерів використовує попередньо навчені ЗНМ для вилучення візуальних характеристик, а потім використовує деякі традиційні методи навчання для визначення місцезнаходження цілі. Такі трекери називаються мережевими трекерами вилучення функцій або трекерами навчання передачі [23].

Кілька архітектур згорткових нейронних мереж продемонстрували значний успіх у класифікації зображень шляхом навчання своїх мереж на великому наборі даних *ImageNet* [24]. Оскільки ці архітектури навчаються за допомогою мільйонів навчальних зразків, згорткові функції цих архітектур мають високу здатність до узагальнення і, отже, можуть бути передані в інші відповідні програми. Кілька систем відстеження використовують можливості узагальнення попередньо навчених ЗНМ. Такі трекери сприймають ці попередньо навчені ЗНМ як екстрактор функцій «чорного ящика», а потім передають згорткові характеристики для моделювання зовнішнього вигляду цілі. Такі трекери продемонстрували [15] кращу точність, ніж мануальні трекери, завдяки хорошій здатності до узагальнення попередньо навчених ЗНМ.

Відомо, що різні рівні архітектури ЗНМ забезпечують різні рівні опису функцій. Тому кілька глибоких трекерів [6] використовують це ієрархічне представлення ознак, вилучаючи ознаки з різних згорткових шарів попередньо навченої ЗНМ. Автори [6] отримали характеристики з кожного згорткового шару мережі *VGG*, а потім навчили кілька слабких трекерів із відповіддю кореляційних фільтрів. Потім вони об'єднуються в більш потужний механізм за допомогою вдосконаленого алгоритму навчання хеджування.

Трекери перевірки функцій (або навчання передачі) розглядають попередньо навчену архітектуру ЗНМ як екстрактор функцій «чорного ящика» та показали кращу продуктивність, ніж мануальні трекери, завдяки широким можливостям



представлення функцій ЗНМ. Незважаючи на те, що ці методи отримують достатньо можливостей узагальнення від попередньо навчених ЗНМ, їх здатність до класифікації є дещо гіршою, оскільки вони не є наскрізними глибокими трекерами. Таким чином, вони не в змозі вловити значні зміни зовнішнього вигляду цілі, і вони піддаються впливу сторонніх елементів.

Трекери *FEN* використовують попередньо навчені ЗНМ для вилучення глибоких характеристик. Незважаючи на те, що вони мають можливість узагальнення, їх продуктивність обмежена через фундаментальну різницю між класифікацією об'єктів і візуальним відстеженням. У той час як класифікація об'єктів має справу з передбаченням міток класів, відстеження об'єктів включає передбачення або визначення місця розташування зразків певного класу. Трекери наскрізної глибокої мережі (*EEN*) долають це обмеження, пропонуючи методи та архітектури глибокого навчання, які спеціально розроблені та навчені для відстеження об'єктів.

Трекери глибокої класифікації розглядають візуальне відстеження як проблему бінарної класифікації, розрізняючи цільові та нецільові області. У цій стратегії дискримінаційна архітектура ЗНМ навчається наскрізним способом, а потім передбачення мережі використовуються для визначення місцезнаходження цілі. На початковому етапі стратегії відстеження глибокої класифікації будується наскрізна архітектура ЗНМ для візуального відстеження. Зазвичай використовується проста архітектура ЗНМ, оскільки бінарна класифікація менш складна, ніж інші проблеми класифікації, такі як класифікація *ImageNet* з 1000 класів. Крім того, оскільки глибші архітектури ЗНМ вимагають великої кількості навчальних зразків, їх повторне налаштування онлайн із доступними обмеженими зразками призведе до перенавчання.

Загальна послідовність процесу відстеження глибокої класифікації показана на рис. 1.3. Тут, на першому етапі відстеження глибокої класифікації архітектура ЗНМ навчається за допомогою великої кількості послідовностей відстеження. У кожній послідовності навколо цілі збирають позитивні та негативні зразки

нецільових регіонів. Мета навчання - вивчити характеристики візуального відстеження, такі як зміна масштабу, деформація, рух розмитість тощо.

За рис. 1.3 (перший етап) архітектура ЗНМ тренується за допомогою великої кількості послідовностей відстеження. Нижче (другий етап): позитивні та негативні зразки збираються з першого кадру послідовності онлайн-відстеження. Потім кілька останніх шарів попередньо навченої ЗНМ повторно налаштовуються (заморожені шари показано синім кольором). І внизу (третій етап): патчі збираються навколо цільової області та оцінюються. Максимальна оцінка передбачення мережі використовується для вибору цільового патча.

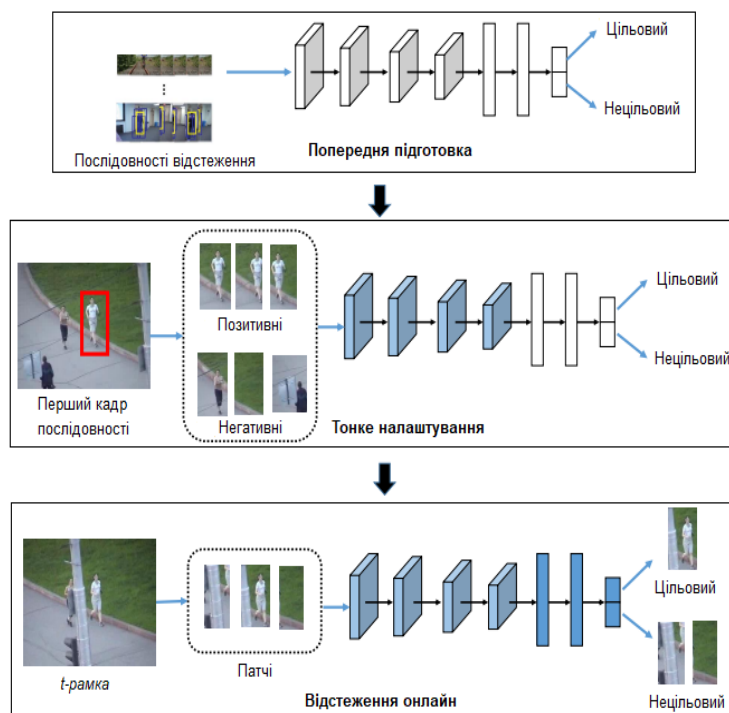


Рис. 1.3. Узагальнений процес відстеження людини за методом глибокої класифікації

Щоб передбачити розташування цілі в інших кадрах, як показано на рис. 1.3 (внизу), патчі збираються навколо останнього відомого розташування цілі. Потім вони оцінюються мережею та отримують їх класифікаційні бали. Патч з найвищим результатом вибирається як цільове місце в цьому кадрі. Щоб адаптувати зміни

зовнішнього вигляду цілі, архітектура ЗНМ безперервно налаштовується протягом послідовностей відстеження через фіксовані інтервали кадрів.

Трекери глибокої класифікації показали високу точність, вивчаючи характерні для цілі дискримінаційні особливості. Ці трекери оновлюють свою модель протягом усієї послідовності відстеження, щоб прийняти зміни зовнішнього вигляду цілі. Незважаючи на те, що такі трекери показали значне покращення точності, обчислювальні витрати на онлайн-оновлення суттєво зменшують швидкість відстеження. Таким чином, ці підходи мало придатні для реальних практичних застосувань.

Відстеження точок траєкторії об'єкта на основі зовнішнього вигляду можна вирішити як проблему відстеження подібності шляхом пошуку подібності відомого цільового шаблону в окремих кадрах. У цій стратегії розташування з найвищим показником подібності використовується для визначення цільового розташування. Група трекерів *EEN* [25] дотримується стратегії відстеження подібності. Трекери глибокої подібності отримують цільовий шаблон із першого кадру відеопослідовності, а потім шукають подібність цього шаблону в наступних кадрах, щоб знайти ціль. Трекери подібності використовують мережеву архітектуру *Siamese*, щоб підвищити потужність узагальнення зіставлення шаблонів, і навчаються на великому наборі даних [23]. Оскільки модель відповідності шаблону подібності не потребує онлайн-оновлення, засоби відстеження подібності досягають швидкості відстеження, що перевищує швидкість відстеження в реальному часі [25].

Трекери глибокої подібності досягли більшої швидкості відстеження в реальному часі, ніж порівнянна точність. Однак їх точність відстеження значно нижча, ніж для трекерів глибокої класифікації, оскільки вони пропускають важливу інформацію про ціль, а отже, їх здатність розпізнавання обмежена. Оскільки ці трекери не збалансовують здатність до узагальнення та розрізнення, такі методи не в змозі відрізнити ціль за допомогою подібних відволікаючих факторів або не адаптуються до часових варіацій візуальних характеристик цілі.

## 1.3 Переваги та недоліки різних підходів

### 1.3.1 Порівняльні набори даних

Візуальне відстеження об'єктів є однією з активних тем дослідження комп'ютерного зору, яка проводиться протягом кількох десятиліть. За цей період було запропоновано велику кількість візуальних трекерів. Важливо оцінити продуктивність цих трекерів, щоб знайти майбутні напрямки їх застосування у задачах відстеження траєкторії. Порівняльні набори даних важливі для візуального відстеження для порівняння продуктивності трекерів. Створено кілька наборів порівняльних даних [15; 21] та вони є загальнодоступними. Кожен набір даних відрізняється від інших кількома факторами, такими як клас цільового об'єкта, розмір, метод анотації, складність (наприклад, щільність натовпу, рівень оклюзії тощо), кут огляду, розташування (в приміщенні чи на вулиці), відстань до камери, статичні та динамічні фони. Крім того, кожен контрольний показник дотримується різних матриць оцінки ефективності.

Набір даних *Caltech Pedestrian* [26] є одним із найбільших і складних наборів даних для відстеження об'єктів. Він містить майже 1000 тисяч кадрів і 2300 унікальних цілей. Відео записується з автомобіля, що рухається. Він містить 350 тисяч обмежувальних рамок і серед них 126 тисяч є закритими. Цей набір даних забезпечує дуже детальний рівень маркування, наприклад частку оклюзії, рамку для оклюзованих частин, масштаб і положення цілей. У цьому наборі даних, хоча 53% цілей закриті в кількох кадрах, 19% закриті в усіх кадрах. З точки зору масштабу, 75% цілей мають розмір від 30 до 80 пікселів, а інші - менше 30 пікселів. На рис. 1.4 показано деякі легкі та складні кадри цього тесту.



Рис. 1.4. Приклади послідовностей тесту *Caltech Pedestrian* [26]

Кілька наборів даних [27] доступні для відстеження одного об'єкта. Однак вони відрізняються один від одного з точки зору схем анотації, ініціалізації та метрики оцінки. Щоб забезпечити справедливу оцінку продуктивності, *OTB* збирає та анотує послідовності з різних наборів даних. *OTB* має 100 загальних послідовностей відстеження об'єктів, і люди є цілями в 72% послідовностей. *OTB* спочатку пропонується з 51 послідовністю відстеження, і вони називаються *OTB2018* [21].

Послідовності в *OTB* анотовані деталями атрибутів, де атрибут відноситься до складного фактора, який впливає на ефективність відстеження. В *OTB* визначено одинадцять атрибутів, таких як зміна освітлення, зміна масштабу, оклюзія, деформація, розмиття руху, швидкий рух, низька роздільна здатність, обертання в площині та поза площиною. *OTB* надає кілька схем оцінки для порівняння продуктивності трекерів. Діаграми точності та успіху використовуються для вимірювання ефективності. На графіках точності середнє значення евклідової відстані між центром земної поверхні та відстежуваною ціллю наноситься на порогове значення 20 пікселів. Коефіцієнт перекриття між наземними і відстежуваними трекерами використовується для ранжирування трекерів у діаграмах успіху. Серед кількох схем оцінювання в *OTB* дослідниками широко використовується метод однопрохідної оцінки (англ. *OPE*). В *OPE* оцінки, трекер ініціалізується в першому кадрі послідовності відстеження, і його продуктивність вимірюється на основі цієї ініціалізації. Оскільки *OTB* є складним завданням,

майже всі візуальні трекари використовували цей тест для оцінки та порівняння своєї продуктивності.

### **1.3.2 Порівняння засобів візуального відстеження об'єктів**

Візуальне відстеження об'єктів проводиться щороку для порівняння продуктивності трекарів без моделі та одного об'єкта. Щороку для цього завдання створюється кілька наборів послідовностей відстеження, які називаються *VOT2020*, *VOT2021* і *VOT2022* [28]. Подібно до *OTB*, кожен контрольний показник ВВО створюється шляхом збору послідовностей з інших наборів даних. Контрольні показники у ВВО використовують точність, надійність і очікуване середнє перекриття як основні показники ефективності. Оскільки тести *OTB* і ВВО будуються шляхом збору послідовностей з інших наборів даних, вони мають кілька загальних послідовностей відстеження.

Класифікація та перегляд трекарів на основі зовнішнього вигляду є складним завданням, оскільки пропонується велика кількість варіантів підходів. Так, періодично виявляються певні прогалини, і в дослідженнях пропонуються майбутні напрямки на основі їх оцінок. Однак, оскільки сучасні методики в ВВО змінюються щороку, більшість висновків і пропозицій досліджень є застарілими. Крім того, майже всі дослідження проводяться для перевірки продуктивності мануальних трекарів, і їх результати можуть бути застарілими після появи глибокого навчання.

Підсумовуючи можна зазначити, що сучасні засоби відстеження об'єктів на основі комп'ютерного зору можна класифікувати за різними категоріями, такими як кількість цілей у послідовності, тип ініціалізації цілі та ін (табл. 1.1). Серед великої кількості засобів у кожній категорії перераховано лише ключові. Залежно від кількості цілей засоби можна класифікувати як одиночні і множинні.

Таблиця 1.1.

Порівняння функціональних характеристик засобів відстеження об'єктів на основі комп'ютерного зору [6; 16; 17; 19]

Метод відстеження	Кількість цілей		Тип інсталяції		Режим процесу		Тип моделі		Особливість	
	Одиночні	Множинні	На основі моделі	Без моделі	Офлайн	Онлайн	Генеративні	Дискримінативні	Мануальні	Глибокі
<i>STRUCK</i>	+	-	-	+	-	+	-	+	+	-
<i>SCM</i>	+	-	-	+	-	+	-	+	+	-
<i>LSK</i>	+	-	-	+	-	+	+	-	+	-
<i>HDT</i>	+	-	-	+	-	+	+	-	-	+
<i>SINT</i>	+	-	-	+	-	+	+	-	-	+
<i>EAST</i>	+	-	-	+	+	-	-	+	-	+
<i>CF2</i>	+	-	-	+	-	+	-	+	-	+
<i>CIT</i>	-	+	+	-	-	+	-	+	+	-
<i>CSK</i>	-	+	-	+	-	+	-	+	+	-
<i>KMS</i>	-	+	-	+	-	+	+	-	+	-
<i>SBT</i>	-	+	-	+	-	+	-	+	+	-
<i>CT</i>	-	+	-	+	-	+	-	+	+	-
<i>IVT</i>	-	+	+	-	-	+	+	-	+	-
<i>DFT</i>	-	+	-	+	-	+	+	-	+	-
<i>ORIA</i>	-	+	+	-	-	+	-	+	+	-
<i>FOGA</i>	-	+	+	-	-	+	-	+	+	-
<i>CRF</i>	-	+	-	+	+	-	+	-	+	-
<i>OLMOAP</i>	-	+	-	+	+	-	-	+	+	-
<i>SGB</i>	-	+	-	+	+	-	+	-	+	-
<i>MDNet</i>	+	-	-	+	-	+	-	+	-	+
<i>MTT</i>	+	-	-	+	-	+	+	-	+	-
<i>MTMVT</i>	+	-	-	+	-	+	+	-	+	-
<i>Branchout</i>	+	-	-	+	-	+	-	+	-	+
<i>SANet</i>	+	-	-	+	-	+	-	+	-	+
<i>SiamFC</i>	+	-	-	+	-	+	-	+	-	+
<i>CNN-SVM</i>	+	-	-	+	-	+	-	+	-	+

Залежно від типу цільової ініціалізації, їх можна класифікувати як на основі моделі і без моделі. Їх також можна класифікувати, як автономні (офлайн) і онлайн

на основі режиму обробки кадрів. Генеративна і дискримінативна є іншою класифікацією, заснованою на типі моделі. За типом функції їх можна класифікувати як трекери ручного налаштування (мануальні) та глибокі трекери. Також розрізняють режим обробки, тип візуальної функції та тип цільової моделі.

Отже, у контексті відстеження траєкторії людини в натовпі, розглянуті у даному розділі методи і засоби мають як переваги, так і недоліки, що наведено нижче.

***Переваги:***

- широка варіативність щодо методів обробки зображень;
- висока продуктивність обробки даних;
- у випадку з методами глибокого навчання, відсутність необхідності втручання оператора у процес виявлення;
- висока точність виявлення, у випадках з методами офлайн обробки даних.

***Недоліки:***

- фрагментарність вихідних даних, щодо актуальної траєкторії людини, що пов'язано із можливою недостатньою кількістю засобів збору візуальної інформації у реальних умовах застосування;
- перевантаженість системи вхідними даними, що пов'язано з обробкою зайвої інформації, яка має бути проаналізованою для виявлення потрібної людини у кадрі нового засобу збору даних (фото/відео-камера) при її переміщенні;
- необхідність узгодження інформації щодо геопросторового розташування засобів збору даних (зображень) для отримання якісних вихідних даних про траєкторію людини.

Таким чином, актуальним є дослідження, що направлене на розробку методу відстеження траєкторії людини, з використанням однієї фото/відео-камери на мобільній платформі, наприклад, автономний транспортний засіб або безпілотний літальний апарат. Це дозволить максимально знизити навантаження на алгоритм обробки зображень за рахунок використання одного датчика, та надасть можливості безперервного контролю за ціллю, що підвищує ефективність процесу формування геопросторової звітності щодо відслідковуваної траєкторії.



## 2 РОЗРОБКА МЕТОДУ ВІДСТЕЖЕННЯ ТРАЄКТОРІЇ ЛЮДИНИ В НАТОВІІ НА ОСНОВІ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ

### 2.1 Побудова досліджуваної математичної моделі

При формуванні обчислювального апарату наголос повинен бути зроблений не лише на обробці конкретного набору даних, а й на модульності системи. Таким чином, має бути можливість зручно замінювати окремі частини обчислювального алгоритму спеціалізованими модулями, адаптованими для конкретних потреб, для можливості розгортання в різних ситуаціях.

Далі наведено основні використовувані інструменти та частини алгоритму, переважно із загальної точки зору. Конкретні використані рішення, їх варіанти та версії будуть описані в окремих підрозділах, присвячених реалізації окремих частин алгоритму роботи системи.

Так, для наочності алгоритм роботи системи описується з певним ступенем абстракції, окремі блоки алгоритму розглядаються в загальних рисах.

Базові блоки алгоритму наведено на рисунку 2.1.

Основні вхідні дані моделі складаються із записаного відео та журналу польотів дрона. Відеозапис надходить в модуль *ImageProvider*, який використовується для обробки зображень. Він завантажує запис по одному кадру, при цьому користувач може, наприклад, вказати певний фрагмент відео, або прискорити роботу програми, пропускаючи кадри тощо. Кожне зображення тут також можна редагувати за потреби. Потім вихідні дані складаються з окремих зображень із мітками часу кадрів відео.

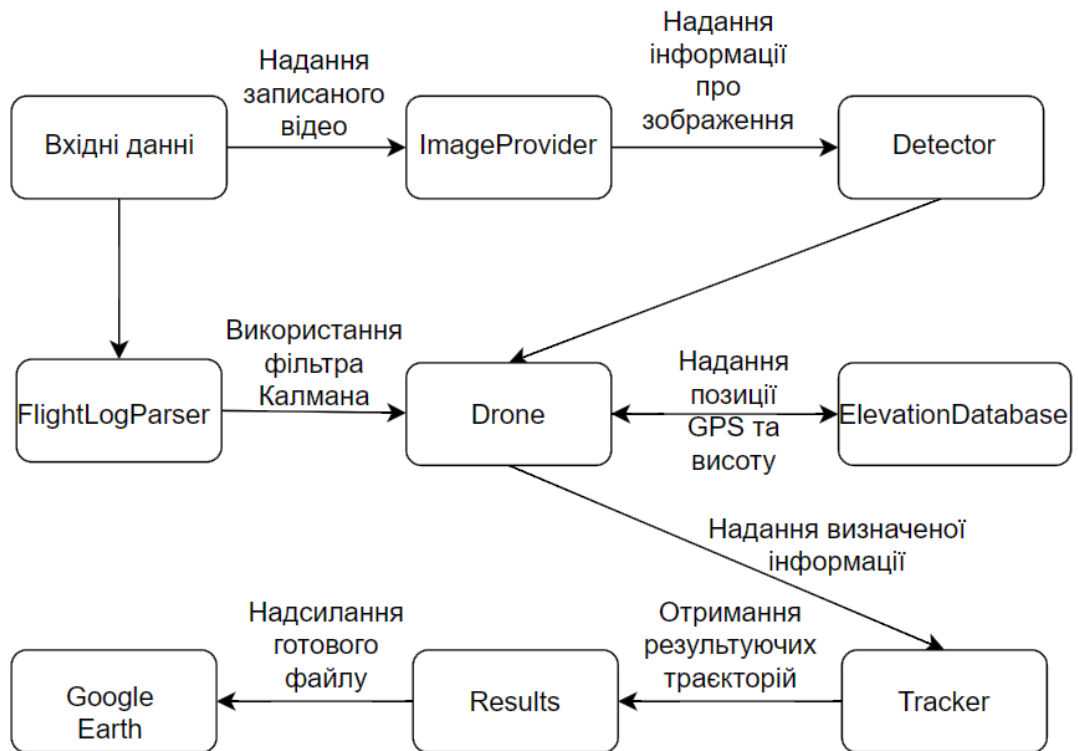


Рис. 2.1. Схема відстеження траєкторії людини у натовпі

Журнал польотів може містити багато інформації, яку потрібно видалити або відредагувати, і її потрібно синхронізувати з часом зйомки відео. Запис відео здійснюється на підвісну камеру дрона (рис. 2.2).



Рис. 2.2. Цифрова камера для дрона

Крім того, дані датчика (камери) можуть бути неточними, тому бажано обробити їх і відфільтрувати помилки. Для цього, наприклад, зазвичай використовується фільтр Калмана. Це передбачено модулем *FlightLogParser*. Потім необхідно шукати осіб на кожному зображенні, яке надає модуль *Detector* для виявлення людей. Через вимогливість середовища (різноманітні об'єкти, кути огляду, відстані, рухомий датчик тощо) необхідно використовувати сучасні рішення на основі згорткових нейронних мереж. Тому можна використовувати інструменти, розглянуті у попередньому розділі, або реалізувати навчання власного детектора, адаптованого до конкретних задач. Результати остаточно формуються в об'єкти, що містять координати зображення окремих людей та їх попередній перегляд на кадрі.

Коли відоме місце розташування людей на зображенні, поворот і положення дрона та його висота польоту, можна визначити місцезнаходження об'єктів у реальному світі, що забезпечується модулем *Drone*. Методи, запропоновані в розділі, використовуються для визначення відстані пішоходів від дрона та обчислення їх *GPS*-координат. Останні фіксуються за допомогою вбудованого у дрон *GPS*-модуля (рис. 2.3).



Рис. 2.3. *GPS*-модуль дрона

Якщо використовується процедура вимагає інших компонентів, вони також повинні бути реалізовані - наприклад, алгоритмом, що спирається на інформацію щодо висоти датчика над поверхнею землі, тому для нього повинна бути база

даних, яка може запитувати висоту для конкретних координат. Таким чином визначається *GPS*-положення людини.

За допомогою попередніх кроків можна отримати наступну інформацію: час зображення, попередній перегляд виявленої людини, її координати на зображенні та положення *GPS*. Завдяки з'єднанню зазначених даних у часі створюється траєкторія відстежуваної особи. Потрібно зазначити, що траєкторія може позначатись як певний відрізок (вектор) маршруту, що, звичайно, не означає весь маршрут, а лише один вектор в певний момент часу.

Для отримання результуючих траєкторій руху осіб необхідно правильно призначити окремі вектори на відеозаписі відповідним особам. Це робиться за допомогою модуля повторної ідентифікації під назвою *Tracker*. Він зберігає записи всіх відомих осіб і список векторів, що їм належать. Він ідентифікує та призначає кожен новий вектор відомій особі або створює нову особу на його основі.

Отже, на цьому етапі є список ідентичностей, де кожна має пов'язаний список треків. Об'єднуючи всі *GPS*-координати з векторів людини, створюється її унікальна траєкторія, яка забезпечується модулем *Results*. Потім отримані маршрути коригуються за допомогою різних методів (наприклад, фільтр Калмана). Потім їх можна експортувати у файл (наприклад, типу *KML*), які можна завантажити та відобразити в *Google Earth* [29].

Весь процес застосування, як описано в попередньому тексті, використовуючи позначення основних модулів та їх зв'язок, можна побачити на рис. 2.1. Можливі розширення (наприклад, бази даних висот) будуть згадані в наступних пунктах, де описано конкретне впровадження різних рішень.

При обчисленні даних, по-перше, необхідно описати різницю між онлайн і офлайн обробкою зображень, оскільки це є ключовим для вибору інструментів.

У випадку онлайн-обробки, це коригування в реальному часі, наприклад, безпосередньо в дроні, який знімає відео. Навіть зі звичайною швидкістю отримання 30 або 60 кадрів на секунду вимоги до швидкості обробки дуже високі, що означає, що багато більш просунутих і вимогливих до обчислень методів виключаються або потрібне спеціалізоване апаратне забезпечення [23].

З іншого боку, офлайн-обробка обробляє відео, записане в минулому. Тому відсутнє обмеження щодо обчислювальної потужності в часі або вимоги є набагато нижчими. Наприклад час обробки одного зображення може становити від секунд до хвилин, не впливаючи на практичне використання програми, тоді як онлайн-обробка займе десятки мілісекунд для тієї ж операції. У цьому випадку кількість використовуваних інструментів не обмежена, і можна використовувати більш складні рішення, такі як більш вимогливі моделі нейронних мереж тощо.

Розроблюваний метод відповідає другому із зазначених варіантів, оскільки базується на використанні записів, виконаних у минулому, тому виконує офлайн-обробку зображень. Іншими словами, немає вимог до надзвичайних обмежень на обчислювальну складність, яка пропонує набагато більше корисних інструментів - особливо машинне навчання, включаючи більш вимогливі штучні нейронні мережі, що важливо з точки зору поточної роботи, оскільки такі методи досягають кращого результату щодо виявлення та відстеження людей [15].

Вибір мови програмування: Python

Більшість сучасних інструментів для виявлення об'єктів або для повторної ідентифікації осіб створені мовою програмування *Python*. У рамках цієї роботи *Python* також використовується, щоб зробити легко розширюваним. Крім того, він також не залежить від платформи - для нього потрібен лише інтерпретатор мови та встановлення використовуваних інструментів [18].

Синтаксис *Python* інтуїтивно зрозумілий і відносно схожий на розмовну мову. На практиці згадані властивості призводять до швидшого та зручнішого (більша гнучкість, менше обмежень) створення додатків. Вони компенсуються нижчою оптимізацією продуктивності, більшими вимогами до пам'яті та вищими вимогами до якості коду для великих проектів, але ці мінуси не є суттєвими для офлайн-обробки зображень. Це робить *Python* одним із найбільш використовуваних інструментів як для початківців, так і для досліджень у сфері машинного навчання.

У поточній роботі використовується кілька зовнішніх бібліотек *Python*. Тут будуть описані найважливіші з них. Для кожної з них може знадобитися

встановлення додаткових пакетів, але всі необхідні залежності можна зручно встановити за допомогою офіційної системи пакування *PyP*. Він завантажує та встановлює компоненти в потрібній версії з онлайн-сховища *PyPI4* [24].

Повний список усіх використаних бібліотек наведено у файлі *requirements.py*, що є звичайною практикою, оскільки тоді можна встановити всі залежності *PyP* за допомогою однієї команди.

Завдяки архітектурі програми та її модульності, її можна запусити зі значно меншою кількістю залежностей. Таким чином, мінімально необхідні бібліотеки для роботи програми наведено у файлі *requirements\_minimal.py*. Використовуючи їх, користувач може уникнути встановлення зайвих пакетів, які не потрібні в даному випадку. Непотрібні розширення не займають місце в пам'яті і не сповільнюють роботу програми. Однак це рекомендовано лише для тестування та налагодження програми.

*NumPy* - це базова бібліотека, яка використовується в науковій діяльності та роботі з великими обсягами даних. Її основу складають *N*-вимірні масиви. Над ними пропонується інтерфейс для важливих операцій, які додатково оптимізовані для продуктивності при роботі з великою кількістю елементів. Багато інших важливих бібліотек (*OpenCV*, *Pandas* тощо) побудовані на цьому пакеті. У даній роботі *NumPy* потрібна як обов'язкова умова. Також вона використовується незалежно в усій програмі, наприклад, для геометричних операцій чи роботи із зображеннями.

*OpenCV* - це структура, орієнтована на комп'ютерний зір. Вона пропонує різноманітні функції від загальної обробки зображень до більш просунутих методів, таких як розпізнавання обличчя. Використовується в проекті для роботи з відео, аналізу та редагування зображень тощо.

*Pandas* - це інструмент для швидкого та ефективного аналізу/обробки даних. На відміну від *NumPy*, його базовим елементом є двовимірна структура, тобто він працює з інформацією, яку можна представити двовимірною таблицею. Він пропонує багато поширених функцій для редагування даних і може зручно

працювати з файлами *CSV*, тому він використовується в програмі для обробки та зберігання журналу польотів.

***TensorFlow*** - це всеосяжна структура, яка підходить для потреб машинного навчання. Цей інструмент дозволяє створювати та навчати штучні нейронні мережі за допомогою зрозумілого інтерфейсу високого рівня тощо. У поточній роботі роботі він служить основою для детектора особи.

***Matplotlib*** пропонує відтворення інтерактивних графіків різних типів, а також їх експорт у файли поширених форматів. Крім того, він може працювати безпосередньо з об'єктами, створеними бібліотекою *NumPy*. У роботі він використовується для візуалізації детальної інформації в тестовому режимі, а також дозволяє відображати траєкторії в попередньому *3D*-перегляді.

***GeographicLib*** пропонує широкий спектр геодезичних операцій, від перетворення координат між різними системами до розрахунків, що працюють із земним тяжінням і магнітним полем. Оригінальна версія реалізована для мови програмування *C++*, але доступний інтерфейс для *Python*, який дозволяє використовувати географічні операції. У роботі передбачено використання з *GPS*-координатами (розрахунки відстаней, переміщень тощо).

***PyTorch*** - це сучасна структура, розроблена для потреб машинного навчання. Це похідна бібліотека *TensorFlow*, але краще інтегрована в *Python* (наприклад, кращі можливості налагодження). У даній роботі використовується нейронна мережа, яка забезпечує повторну ідентифікацію осіб.

#### *Додаткові інструменти*

***Google Earth*** - це програма, яка дозволяє переглядати віртуальний глобус. Він фіксує реальну форму планети за допомогою супутникових зображень, а також пропонує *3D*-зображення поверхні. У даній роботі вона використовується для відображення отриманих траєкторій відстежуваних осіб на реальній карті.

***Open Elevation*** - це інструмент із відкритим кодом, який дозволяє створювати сервер із даними про висоту. Потім можна спілкуватися з ним за допомогою *http*-запитів, коли він відповідає на надіслані запити, що містять положення *GPS* із значеннями висоти. Однак сам *Open Elevation* - це лише інструмент, який не

зберігає жодних даних. Він отримує інформацію з файлів *GeoTIFF*, які містять необхідні дані.

***GeoTIFF*** - це формат даних, призначений для зберігання та передачі геодезичних даних. Він заснований на стандартному форматі *TIFF*, який використовується для роботи з растровою графікою. Однак він додає метадані, які вказують, наприклад, на просторове розташування, дату або систему координат. Він зазвичай використовується для зберігання інформації про висоту над рівнем моря або іншої інформації про земну поверхню, яка може бути відображена завдяки типу формату та стилю зображення (наприклад, з інтенсивністю кольору відповідно до висоти).

***GDAL*** - це консольний інструмент для роботи та редагування файлів, що містять геопросторову інформацію (наприклад, згаданий *GeoTIFF*). Це дозволяє їх аналізувати, редагувати, конвертувати між векторними та растровими типами тощо. У роботі він використовується, наприклад, для вилучення області інтересу з наборів даних висот.

***QGIS*** - це геоінформаційна система, яка дозволяє візуалізувати, редагувати та працювати з картами, моделями поверхні та іншими даними в різноманітних форматах у зрозумілому графічному інтерфейсі. При використанні цифрової моделі висоти тут можна проаналізувати, наприклад, профіль висоти зони моніторингу. На відміну від звичайних програм ГІС, які часто є дорогими, *QGIS* базується на принципі відкритого коду та доступний безкоштовно.

***Shell skripty***: у проекті є кілька сценаріїв оболонки для редагування відео (наприклад, зміни роздільної здатності або видалення звукової доріжки). Однак вони не використовуються користувачем під час звичайної роботи з програмою, вони використовуються лише для полегшення ручної попередньої обробки відео, якщо це необхідно.

***PyCharm***: додаток реалізовано в середовищі розробки *PyCharm*. Він пропонує широкий спектр функцій підтримки, перевірок коду та спеціальних розширень, що допомагає підтримувати ясність і якість вихідного коду. Він також пропонує повну версію безкоштовно для навчальних цілей.



## 2.2 Розробка методу відстеження траєкторії людини в натовпі за допомогою технологій комп'ютерного зору

У цій частині роботи більш детально описано обробку вхідних даних та їх модифікацію. Вхідні дані складаються з двох ключових блоків – відеозапису та журналу польотів дрона. Ці елементи має надати користувач під час запуску програми. Зображення поступово завантажуються із запису, який за потреби може пройти додаткову попередню обробку. Журнал польотів створює дрон під час зйомки відео. У кожному пристрої такий вихід може відрізнятись, тому його необхідно обробити та витягти з нього лише важливу інформацію, необхідну для визначення місцезнаходження осіб, які відстежуються.

Зображення завантажуються та обробляються модулем *ImageProvider*. Його завдання - надати окремі зображення разом із міткою часу кадру. Це використовується для обчислення *GPS*-положення людини, коли необхідно знати дані про дрон і його орієнтацію в даний момент. Його також можна використовувати для покращення повторної ідентифікації під час відстеження людей. Згодом він застосовується в кінці всієї програми при генерації траєкторій відстежуваних осіб і наведенні напрямку їх руху.

До кожного завантаженого зображення можна застосувати різні налаштування. Вони також виконуються у згаданому модулі, але вони надаються об'єктом *ImageTweaker*. Тому вони не реалізовані безпосередньо в окремих варіантах *ImageProvider*. Це виконується для того, щоб методи були доступні через прямий *API ImageTweaker* із можливістю налаштувати його параметрів та порядку виконання операцій.

*ImageProvider* реалізований таким чином, що надається можливість відкладеного оцінювання, тобто окремі елементи завантажуються лише тоді, коли вони потрібні. Завдяки цьому неактивні знімки не займають місце в пам'яті, і представляється потенційно нескінченна структура даних. Конкретні варіанти *ImageProvider* та їхні параметри описані нижче.

*ImageProviderFromVideo* завантажує кадри з відеофайлу. Для зручності користувача пропонується можливість визначення конкретної ділянки відео за допомогою індексу початкового та кінцевого кадрів. Оскільки відео зазвичай знімають із частотою кадрів 30 або 60 кадрів на секунду, модуль також допускає можливість пропуску кадрів, тобто користувач визначає, скільки кадрів ігнорувати, перш ніж один буде повернено. Таким чином, додаток можна значно прискорити, якщо достатньо оцінити, наприклад, лише одне зображення в секунду. Цей модуль призначений для використання в програмі в стандартному режимі, і при нормальному використанні не має бути потреби в інших варіантах *ImageProvider*. Однак інші реалізації доступні для тестування та налагодження.

*StaticImageProvider* і *RandomImageProvider* модулі призначені лише для налагодження/тестування. Вони працюють за схожим принципом. Вони можуть встановити максимальну кількість кадрів, яку вони можуть надати, а також кількість кадрів в секунду, за допомогою якої додаються часові мітки окремих знімків. Потім *StaticImageProvider* завжди повертає те саме попередньо встановлене зображення, тоді як *RandomImageProvider* щоразу генерує нове зображення з випадковими значеннями пікселів і необхідною роздільною здатністю.

Зображення, зняте камерою, не є досконалим і може містити, наприклад, шум, артефакти, викликані освітленням або іншими високочастотними компонентами. Тому редагування зображень є звичайною практикою комп'ютерного зору. Мета полягає в тому, щоб усунути зазначені дефекти, але при цьому зберегти важливі ознаки, за допомогою яких потім буде вилучена необхідна інформація (наприклад, під час виявлення та повторної ідентифікації осіб).

За допомогою класу *ImageTweaker* до кожного завантаженого зображення можна застосувати різні методи налаштування. Це дозволяє попередньо встановити бажані операції в будь-якому порядку та з додатковими параметрами. Деякі із запропонованих модифікацій описано нижче.

Фільтр Гауса (*Gauss filtr*) є одним із найбільш використовуваних методів для видалення шуму на зображенні. Він працює шляхом поступового застосування

згортки з маскою до всього зображення, тоді як її розмір вказує, наскільки широка область навколо кожного пікселя буде зачеплена. Потім значення полів маски визначаються функцією Гауса [8]. Результат застосування фільтра Гауса візуально виглядає як розмиття зображення та згладжування пікселів, які не відповідають навколишньому середовищу. Водночас, краї також розмиті (відповідно до розміру маски), що може бути небажаним. Тому доцільно розглянути можливість використання методу залежно від ситуації. Наприклад, якщо виявлення країв буде використовуватися далі, але загалом не повинно бути проблем під час обробки зображення за допомогою нейронних мереж.

Білатеральний фільтр (*Bilateral filter*) [8] також використовується для видалення шуму на зображенні. Однак, на відміну від більшості інших часто використовуваних фільтрів, він може зберігати інформацію для подальшого виділення меж навіть у кольорових зображеннях, щоб відобразити сприйняття кольорів і меж людським оком. Він використовує принцип фільтра Гауса, але також враховує взаємні відмінності в інтенсивності пікселів. Через це він вимагає більше обчислень.

Вирівнювання гістограми використовується для покращення контрастності зображення. Однак зміни розраховуються на основі гістограми всього зображення в цілому, що має покращити якість зображення загалом, але може призвести до негативних змін у деяких його частинах.

Таким чином, у програмі реалізовано метод адаптивного вирівнювання гістограми, який обробляє зображення меншими частинами та застосовує зміни локально, де це необхідно. У результаті, наприклад, темні ділянки повинні бути виявлені, а деталі відновлені в надто яскравих місцях, де освітлення видалило важливі елементи. Однак у місцях, де виникає шум, він може бути посилений. Алгоритм *CLAHE* (контрастно обмежене адаптивне вирівнювання гістограми) намагається запобігти цьому, що обмежує екстремальні значення контрасту.

Обробка польотного журналу забезпечується модулем типу *FlightLogParser*, який на вхід отримує інформацію в тому вигляді, в якому вона записана дроном. У даному випадку це дрон *DJI Spark* (рис. 2.5).



Рис. 2.5. Дрон *DJI Spark*

Так, реалізовано лише один конкретний варіант, а саме *FlightLogFarserDjiSpark*, який описано далі. Оскільки журнал польоту містить дані всього польоту, але для робочих цілей потрібні лише дані під час запису камерою, першим кроком є розділення його на підчастини, що відповідають відеозаписам. Це виконується атрибутом камери, який вказує, чи активна вона під час зйомки. Оскільки необроблені дані містять багато зайвої інформації та можливих помилок, частини запису далі перетворюються на попередньо визначену форму, де результатом є файл *CSV* із чітко визначеними назвами стовпців і використаними одиницями. Усі дані без мітки часу видаляються.

У рамках поточної роботи досліджується кілька рішень для виявлення людини за допомогою згорткових нейронних мереж. До модулю під назвою *Detector* зображення входить разом із міткою часу. Вихід - список попередніх переглядів знайдених осіб. Взаємодія з модулем забезпечується чітко визначеним інтерфейсом, а саме методами *detect* і *detect\_multi*. Перший з них обробляє окремі зображення, а другий може вміщувати декілька з них одночасно. Обидві функції дають однаковий результат, але деякі детектори можуть дозволяти паралельну обробку кількох зображень одночасно, і ця процедура може значно пришвидшити обробку даних. Потім вигляд виявленої особи представляється об'єктом *PersonView*. Він зберігає координати обмежувальної рамки (прямокутника, що обмежує знайдений об'єкт) і ділянки кадру з відповідною людиною, пропонуючи

кілька допоміжних функцій над ними. Для нейронних мереж, результатом яких є маска знайденого об'єкта замість обмежувальної рамки, спочатку необхідно обчислити обмежувальний прямокутник для маски.

Схема алгоритму виявлення осіб на зображенні наведена на рис. 2.6.

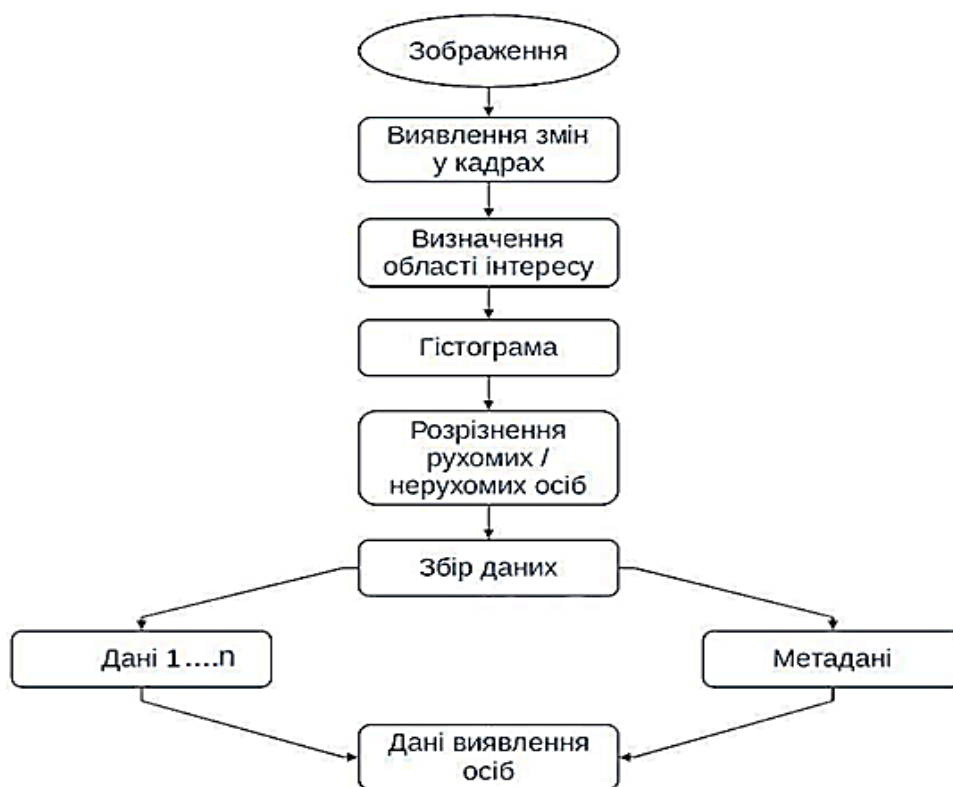


Рис. 2.6. Схема алгоритму виявлення осіб на зображенні

### *Використовувані детектори осіб*

Поточна реалізація пропонує кілька варіантів детекторів, заснованих на принципі згорткових нейронних мереж. Завдяки модульності алгоритму можливе підключення додаткових модулів. Тому, якщо цього вимагає конкретна ситуація використання, можна навчити спеціалізовані детектори, адаптовані до даних потреб. Однак розробка та навчання нейронної мережі для виявлення осіб не входить у сферу поточного дослідження, тому використовуються загальнодоступні рішення, які можна модифікувати для потреб даної роботи. Реалізовані модулі описані нижче.

*OpenPoseBinaryDetector* заснований на проекті *OpenPose*, який шукає та з'єднує зображення окремих частин людського тіла. Тут підтримуються як *CPU*,

так і *GPU*, а також різні версії моделей тіла (різні частини тіла, обличчя тощо). Вибір типу виявлення можна налаштувати в модулі.

Позитивною рисою зазначеного модуля є відсутність помилкових спрацьовувань (виникають вкрай рідко). Таким чином, якщо кілька точок виявляються в дублікаті з нижчим рівнем достовірності, їх майже ніколи не буває достатньо для створення цілої дійсної особи, або використовуються як частина дійсного виявлення. Ця властивість компенсується дуже високими вимогами до обчислювальної потужності. Наприклад, при відстеженні більш віддалених осіб, коли виявлення потрібно виконати на зображенні з високою роздільною здатністю, цей детектор не є ідеальним з точки зору вимог до обчислення та пам'яті.

Слід зазначити, що можна встановити роздільну здатність зображення, яку *OpenPoseBinaryDetector* використовуватиме під час самого виявлення. Оскільки вимоги до обробки зображення в оригінальній роздільній здатності можуть бути надзвичайними саме через вищезазначені вимоги до обчислення та пам'яті, цей параметр можна змінити спеціально для потреб модуля, не впливаючи на вихідне зображення. У результаті координати виявлених осіб масштабуються до вихідного зображення, тому зниження якості зображення обмежить лише цю частину програми, а не інші (наприклад, повторна ідентифікація може відбутися знову в оригінальній якості). Крім того, реалізовано обмеження щодо мінімальної кількості частин тіла, необхідних для успішного пошуку виявленої людини, щоб не відновлювати кожного разу іншу частину людини (наприклад, голова на одному зображенні та вся особа на іншому). Існує також мінімальне значення надійності виявлення, якого необхідно досягти, щоб розпізнати дійсну частину тіла. Це значення може бути задовільним, оскільки детектор не має надто багато проблем із повторними виявленнями саме через вимогу щодо мінімальної кількості виявлених частин, що досить ефективно запобігає хибнопозитивним результатам.

У рамках проекту *TensorFlow* є пропозиція моделей нейронних мереж, навчених на різних наборах даних. *TensorFlowV1ModelDetector* дозволяє використовувати будь-які з тих, які підтримують виявлення особи. Перелік всіх представлених моделей і необхідну інформацію можна за відповідним джерелом

[21]. Дані окремих моделей експортуються у файл, який містить як структуру даної мережі, так і вивчені значення її параметрів.

У рамках згаданого проекту були протестовані детектори, навчені на наборі даних *COCO24*, які мають висновок обмежувальної рамки, наприклад «*SSD Resnet 50 v1 FPN*» з функцією *extractor*, «*Faster R-CNN Inception v2*», що є конфігурацією для набору даних *MSCOCO* [26]. Тут перевагою над *OpenPoseBinaryDetector* є узгодженість обмежувальних прямокутників із зображеннями людей. Моделі із *TensorFlow* також повинні досягти значно швидшого часу виявлення (вони шукають не десятки точок на тілі, а людину в цілому).

В рамках *TensorFlowV1ModelZooDetector* можна налаштувати рівень надійності виявлення нейронної мережі (мережа вказує його для кожного потенційного результату), якого необхідно досягти, щоб знайдений об'єкт вважався дійсним. Нижче значення означає правильну ідентифікацію більшості осіб, але може виникнути помилкове виявлення або дублювання кількох обмежувальних рамок для однієї особи. Більше значення означає менше хибних виявлень, але в той же час збільшується кількість проігнорованих об'єктів, які в іншому випадку були б дійсними (наприклад, люди, розташовані далі або неналежно освітлена).

Алгоритм отримання виявлених осіб використовується в модулі *TensorFlowV1ModelZooDetector*.

Змінна вказує на тип виявленого об'єкта для людини, є мінімальним заданим значенням достовірності виявлення, це максимально допустиме перекриття двох обмежувальних рамок:

```

obtain all detected objects from the neural network
for every  $o \in \text{objects}$  do
    if (type of  $o == \text{PERSON}$ ) and (confidence of  $o \geq \text{THRESHOLD}$ ) then
        append  $o$  to candidates
    end if
end for
for every  $x \in \text{candidates}$  do
    for every  $y \in \text{candidates}$  do
        if (bounding box overlap of  $x$  by  $y \geq \text{TOLERANCE}$ ) and ( $x \neq y$ ) then
            if confidence of  $x \leq \text{confidence of } y$  then

```

```

        delete x from candidates
    else
        delete y from candidates
    end if
end if
end for
end for
return candidate

```

Для часткового вирішення проблеми дублювання результатів реалізована можливість встановлення максимально допустимого збігу осіб. Якщо дві (чи більше) обмежувальні рамки перекриваються більше, ніж дозволяє обмеження (наприклад, більше половини площі), розпізнається лише результат із вищим значенням достовірності, а інший ігнорується.

*DetectorUsingPredefinedSerializedResults* дозволяє завантажувати результати попередньо підготовленого виявлення особи з диска, таким чином обходячи потребу повторювати реальне виявлення під час кожного запуску програми. Результати можна попередньо згенерувати та зберегти на диску. Під час роботи програми результати можна лише завантажувати, що значно прискорює програму порівняно з реальним виявленням за допомогою нейронних мереж. Однак ця процедура більше призначена для тестування та налагодження.

Зображення обробляються програмою одне за одним. Тому під час роботи програми неможливо спочатку виявити людей на всіх знімках, а потім працювати з ними далі (надмірні вимоги до оперативної пам'яті, необхідно буде обмежити максимальну кількість кадрів). Таким чином, якби користувач хотів спробувати різні варіанти налаштувань і оптимізацій (наприклад, під час відстеження людей), йому довелося б кожного разу виконувати виявлення особи знову, що вимагає додаткових обчислень. Тому слід застосувати модуль *DetectionsExporter*, який дозволяє виконувати окреме виявлення на вказаному відео за допомогою вибраного детектора особи. Результати потім завантажуються на диск, де файл, що містить серіалізовані виявлення, створюється для кожного кадру відповідно до мітки часу. При завантаженні зображень необхідно звернути увагу на те, що мітки часу можуть дещо відрізнятись в різних версіях бібліотек *NumPy* або *OpenCV*.



Назви отриманих файлів можуть не збігатися в точності, і їх необхідно завантажувати з допуском, розмір якого повинен залежати від частоти кадрів запису.

Методи визначення *GPS*-положення людей, як вихідні дані використовують *GPS*-положення, яке містить широту, довготу та висоту. Для розрахунку положення відстежуваних осіб модуль може використовувати всі дані, отримані на попередніх кроках програми. Таким чином, це журнал польотів, з якого можна визначити орієнтацію дрона в просторі (значення вибираються на основі мітки часу зображення), і вхідне зображення зі знайденими особами та їх координатами.

Орієнтацію дрона та камери в просторі (напрямок огляду) можна описати *3D*-вектором у системі координат «Північ-Схід-Вниз» (англ. *NED*), яка зазвичай використовується в авіації. Його вісь *x* вказує на північ, *y* - на схід і вниз до поверхні Землі (більшість подій відбувається під ешеленом польоту, тому визначення позитивного зростання координат *y* цьому напрямку більш придатне для використання на практиці). Для простоти, орієнтація дрона та камери розглядатиметься як однакові дані. Вирішення цього питання виконується модулем *FlightLog*.

У всіх запропонованих методах також необхідно визначити напрямок, в якому лежить ціль особи, тобто тривимірний напрямний вектор півпрямой. Однак для цього потрібні не тільки координати зображення виявленого об'єкта, а й інформація про поле зору записуючого пристрою. Для цього створено базовий модуль *Camera*.

Для отримання більш точного результату при визначенні положення людини в просторі також може допомогти інформація про висоту на місцевості, для чого використовується модуль *ElevationDatabase*, що дозволяє отримати доступ до таких даних.

Крім іншого, дрон також може використовувати інші модулі (наприклад, при отриманні та редагуванні результатів, в яких також може відобразитися його траєкторія). Тому пропонує кілька обов'язкових атрибутів для більш зручного пошуку інформації, таких як ім'я, позиція, висота польоту тощо. Також корисними

можуть бути такі методи, як *update\_to\_time*, який дозволяє синхронізувати всі внутрішні властивості модуля з потрібним моментом часу.

Простого 3D-вектора огляду дрона у просторі недостатньо для обчислення GPS-положення людини. Особи можуть з'являтися в будь-якій точці зображення, тому необхідно визначати напрямний вектор півпрямой (напрямок людини).

Модуль *Camera* використовується для визначення вектора напрямку руху людини. Він зберігає дані про висоту пікселів і ширину зображення, які можна отримати з відеозапису, але головним чином інформацію про розмір поля зору камери (горизонтальний і вертикальний *FoV* у градусах), що характерно для конкретного тип записуючого пристрою та потребує налаштування. Так розглядається варіант *CameraDjiSpark*, який описує параметри камери (див. рис. 2.2), що передбачені стандартним обладнанням дрона *DJI Spark*.

Якщо виробник не вказує кут огляду камери, ці дані можна визначити за допомогою бібліотеки *OpenCV* і калібрувальних вимірювань. Якщо відомі лише горизонтальні або вертикальні складові, другу частину можна розрахувати за допомогою методів *calculate\_horizontal\_fov* (2.1) і *calculate\_vertical\_fov* (2.2).

Це базується на співвідношеннях, наведених у наступних рівняннях.

Розрахунок висоти зображення (2.1)

$$\tan\left(\frac{fov_{horizontal}}{2}\right) = \frac{width}{distance}; \quad (2.1)$$

Розрахунок ширини зображення (2.2)

$$\tan\left(\frac{fov_{vertical}}{2}\right) = \frac{height}{distance}; \quad (2.2)$$

Розрахунок ширини/висоти зрізу поля зору камери (2.3)

$$ratio = \frac{width}{height}; \quad (2.3)$$

де «*width / height*» вказує ширину/висоту видимого зрізу, яка змінюється залежно від відстані камери до об'єкта; горизонтальна (*horizontal*) / вертикальна

(*wertical*) складова поля зору позначається змінною, а потім виражає співвідношення сторін зображення.

Фактичний розрахунок напрямного вектора напівлінії відбувається в два етапи. У першому обчислюється різниця між вектором напрямку огляду дрона та вектором напрямку людини. Він розділений на вертикальний (2.4) і горизонтальний (2.5) компоненти (в орієнтації згідно з *NED*), визначені за допомогою наступних рівнянь.

Обчислення різниці між вектором напрямку огляду камери та вектором напрямку людини, вертикальний (2.4)

$$angle_{horizontal} = \frac{person_x - center_x}{width \cdot fov_{horizontal}}; \quad (2.4)$$

Обчислення різниці між вектором напрямку огляду камери та вектором напрямку людини, горизонтальний (2.5)

$$angle_{horizontal} = \frac{person_y - center_y}{height \cdot fov_{vertical}}; \quad (2.5)$$

де змінна «*angle*» вказує на кут відхилення напрямного вектора людини. Ця змінна вказує на координати центру зображення (уявну точку), де знаходиться людина (наприклад, центр обмежувальної рамки або її нижній край як припущення про місце, де стоїть особа).

Другим кроком є остаточний розрахунок тривимірного вектора напрямку для даної людини в системі координат дрона.

Остаточний розрахунок тривимірного вектора напрямку для окремої людини в системі координат дрона (2.6)

$$\vec{v} = distance \cdot [1, \tan(angle_{horizontal}), \tan(angle_{vertical})]. \quad (2.6)$$

Тут напрямок осі відповідає *NED*, тобто вісь пролягає у напрямку огляду дрона, проходячи через корпус дрона зверху вниз.

*Визначення положення об'єкта за допомогою даних щодо висоти розташування камери*

Цей метод базується на використанні інформації про висоту поверхні, по якій пересуваються люди. Тому для його функціонування потрібен модуль типу *ElevationDatabase*, за допомогою якого він приблизно визначає положення людини з необхідною точністю.

Алгоритм вищого рівня визначення положення людини включає отримання *3D*-положення і орієнтацію дрона з *FlightLog* та має вигляд:

```

get 3D position and orientation of drone from FlightLog by frame timestamp
calculate drone elevation using its position and ElevationDatabase
if drone elevation unknown then
    return null
end if
get person direction vector from Camera
transfer person direction vector from drone-coordinates to NED-coordinates
approximate closest ground point in person direction using elevation database
if approximation failed then
    return null
else
    return approximated ground position as person position
end if

```

Далі необхідно отримати дані з журналу польотів із зазначенням розташування дрона в *3D*-просторі, зокрема положення (висота над рельєфом, широта і довгота) і обертання (рискання, тангаж, крен). Згодом висота точки, над якою завис дрон, отримується з *ElevationDatabase*. Якщо вона невідома, алгоритм не може продовжувати роботу (позиція не повертається), інакше до неї додається висота дрона над місцевістю для визначення висоти самого дрона.

Оскільки висота дрона визначається відносно поверхні і півлінії на якій знаходиться відстежувана особа, для кінцевої широти та довготи не має значення, якщо загальні дані розташовані, наприклад, на 100 м нижче/вище від реальної земної поверхні. Таким чином, якщо дана модель дрона також пропонує

інформацію про висоту, все одно краще розрахувати її відносно використовуваної бази даних.

Далі за допомогою модуля *Camera* отримується напрямний вектор півлінії, який відображається в системі координат дрона (вісь пролягає в напрямку огляду, проходить через дрон зверху вниз). Щоб визначити *GPS*-координати, їх потрібно спочатку перетворити в систему координат «Північ-Схід-Вниз». Для цієї операції використовується відповідна матриця перетворення [29], де отримується напрямок (починаючи з положення дрона), у якому перебуває людина. Єдиною інформацією, якої не вистачає для обчислення *GPS*-положення людини, є відстань до об'єкта. Це можна виміряти або оцінити за допомогою спеціальних приладів іншими методами. Однак досліджуваний метод базується на припущенні, що дрон не має подібних спеціалізованих пристроїв, тому згадану відстань необхідно приблизно визначити, використовуючи знання висоти поверхні (на якій очікується пересування осіб).

#### *Приблизне визначення положення за висотою*

Апроксимація найближчої точки земної поверхні в заданому напрямку є вектором напрямку людини, або максимальною відстанню від дрона, на якій може перебувати людина. З міркувань оптимізації реалізовано пакетний пошук простору, щоб усі необхідні значення не обчислювалися окремо, а для кожного окремого кроку виконувалось звернення до *ElevationDatabase*. *DroneDjiSpark* також дозволяє встановлювати додаткову максимальну горизонтальну відстань між виявленою людиною та *GPS*-позицією дрона, наприклад, щоб обмежити небажані елементи поблизу.

Алгоритм для найближчої точки, в якій вектор напрямку огляду дрона перетинає поверхню Землі має наступний вигляд:

```
while desiredAccuracy was not reached do
  do steps in the direction v → person from pos0 until maxDist is reached
  find the first position posi below ground
  if stepSize ≤ desiredAccuracy then
    if posi was found then
      return posi
```

```

        else
            return null
        end if
    end if
    previousStepSize = stepSize
    stepSize = stepReductionCoef * stepSize
    if posi was found then
        pos0 = posi-1
        maxDist = previousStepSize
    end if
end while

```

### *Відстеження траєкторії з повторною ідентифікацією особи*

Далі описано реалізацію, яка забезпечує розрізнення окремих знайдених осіб, створення ідентифікаторів і присвоєння цим ідентифікаторам нових знахідок. Це забезпечується модулем *Tracker*, який зберігає відомі особи та призначає їм нові вектори осіб.

Вектор людини у часі (клас *PersonTimeFrame*) складається з мітки часу кадру, в якому вона була виявлена, обмежувальну рамку, що розмежовує його місце в початковій точці (шляхом фотографування), попереднього перегляду цієї особи (вирізання зображення) та її *GPS*-положення. Для оптимізації програми вона також може містити додаткові необов'язкові дані, наприклад, набір раніше вилучених ознак, які потім не потрібно перераховувати при порівнянні з новоприбулими особами.

Основи процедур, що використовуються для повторної ідентифікації осіб, реалізовано на основі двох функціональних модулів, а саме *HistogramTracker* і *AlignedReIdTracker*. Ці модулі можна використовувати для повторної ідентифікації без контексту на основі введених зображень окремих осіб.

Положення на основі координат зображення - порівнювані особи повинні бути розташовані поблизу нової траєкторії на основі положення обмежувальних рамок, визначених детектором осіб. Якщо їх відстань перевищує максимально допустиме значення, вони ігноруються. Дозволена відстань розраховується відповідно до введеної максимально дозволених швидкості людини та допустимого

відхилення. Цей метод сприйнятливий до різких змін в камері, які можуть призвести до хибнонегативного результату.

Розташування за координатами *GPS* - процедура подібна до обмежень за координатами зображення, але тут використовується визначене місцеположення людей за *GPS*. Це призводить до усунення чутливості до різких рухів камери, оскільки положення *GPS* має визначатися незалежно від обертання дрона. Проблема полягає в неточності, яку вносять інші частини алгоритму (наприклад, журнал польоту або модель висоти на поверхні).

*AlignedReIdTracker* модуль базується на нейронній мережі *AlignedReID* [5]. Зокрема, реалізовано використання вільнодоступної моделі, побудованій на архітектурі *Resnet50*. Вона використовується як так звана магістраль, яка є типом структури однієї частини згорткової нейронної мережі, навченої на наборі даних *Market-1501* [19]. Дана мережа навчена за допомогою комбінації глобальних і локальних характеристик, отриманих із зображень людини [4]. Локальні функції, які належним чином відкореговані для відповідності між порівнюваними зображеннями, допомагають належним чином навчити мережу. Однак для підсумкового порівняння вже використовуються суто глобальні ознаки.

*HistogramTracker* модуль заснований на базовій техніці аналізу зображень. Виділення ознак відбувається шляхом обчислення багатовимірної гістограми. Для кожного кольорового компонента зображення визначається кількість бінів (інтервалів для різної інтенсивності кольору), у межах яких підраховується кількість відповідних пікселів. Отримані значення нормалізуються, а потім об'єднуються у вектор довжини. Цей прийом може виявитись більш ефективним, ніж просто порівняння окремих компонентів кольору. Крім того, гістограми не порівнюються за допомогою евклідової відстані. Однак, сама метрика може давати суперечливі результати. Тому реалізовано 2 різні показники. Перший - це перетин гістограм, який обчислюється з урахуванням кількості пікселів для певної комбінації кольорів і вказує загальну кількість комбінацій на гістограмі. Другий метод - відстань Хеллінгера. Обидва використовувані методи працюють по-різному. У той час як більш високе значення перетину гістограм (параметр *intersect\_threshold*) означає

більшу подібність, з методом Хеллінгера (параметр *hellinger\_threshold*) це прямо протилежне. Тут менше значення вказує на більшу подібність. Результиуюча траєкторія призначається лише в тому випадку, якщо обидва методи досягають необхідної межі подібності й водночас максимальної подібності для однієї виявленої особи.

Результатом роботи програми є траєкторії відстежуваних осіб, визначені їх *GPS*-положенням у часі. Результати мають відобразитися в *Google Earth*. Усю обробку та зберігання результатів виконує клас *Results*. Для наочності результатів була введена можливість збереження також координат дрона, які отримані безпосередньо з бортового журналу.

Цей клас реалізує опції для експорту результатів у файли *GPX*, що є загальноживаним стандартом для зберігання та передачі записів *GPS*. Однак він пропонує лише обмежену підтримку в *Google Earth*, оскільки це не рідний формат цієї програми. Тому також реалізовано експорт у файли типу *KML*. Обидва типи файлів і те, як зберігати координати, описано далі. Для їх створення використовується стандартна бібліотека *Python* для роботи з форматом *XML*.

Коригувати відстерення траєкторії людей можна кількома способами. Перший - видалення осіб, які не досягають мінімально обраної кількості треків. Другий - видалення осіб, які з'явилися менше ніж мінімально визначений час (наприклад, кілька секунд). Так, вектори однієї людини можна агрегувати за часовим інтервалом - для позицій *GPS* за певний час значення широти та довготи. Нові вектори, створені таким чином, потім замінюють початкову траєкторію, що може усунути надмірну сегментацію маршруту.

Фільтр Калмана також можна застосувати до *GPS*-положень отриманих векторів, щоб згладити траєкторію. Отримане відображення потім керується програмним шляхом за допомогою системного виклику, коли шлях до файлу *KML* передається як параметр програмі *Google Earth*. Однак це неможливо зробити з файлом *GPX*, який *Google Earth* дозволяє відкривати лише вручну. У цьому випадку або якщо вказано недійсний шлях *Google Earth*, файл відкривається в програмі за замовчуванням, пов'язаний із цим форматом.



Формат *GPX* [29] було створено з метою уніфікації та спрощення обміну даними між географічними додатками. Його поточна версія була створена в 2004 році і побудована на схемі *XML*, який є добре відомим і широко поширеним стандартом для зберігання та передачі даних у незалежний від платформи спосіб. Це полегшує роботу з файлами *GPX*, тому їх використовують сотні програм і веб-додатків (наприклад, *Google Earth*). Файл *GPX* може складатися з різних структурних елементів, які зазвичай містять дані *GPS* та іншу додаткову інформацію. Наприклад, точка *wptType* представляє будь-яку точку інтересу. Так, *rteType* символізує будь-який маршрут, що складається із згаданих елементів тощо. Якщо попередньо визначені типи не відповідають потребам, можна розширити формат за допомогою власних доповнень.

Елемент *trkType* виявляється найбільш придатним для запису траєкторій людей. Він складається з сегментів *trksegType*. Їх можна використовувати для розділення ділянок траєкторії, наприклад, у випадках, коли пристрій *GPS* втратив і знову знайшов сигнал. Сегменти зберігають шлях як упорядкований список точок *wptType*, які містять довготу та широту. Крім того, вони можуть містити іншу інформацію, наприклад, висоту, назву та опис локації тощо.

Стандарт *KML* був створений як базовий для програми *Google Earth* з метою передачі не лише географічної інформації, але й додаткових аспектів, наприклад, різноманітних геометричних форм і візуальних стилів. Завдяки цьому він може передавати *GPS*-маршрути, креслення та багато іншого. Подібно до формату *GPX*, *KML* базується на стандарті *XML*. Тоді формат *KMZ* представляє стиснуту версію файлу *KML*.

У свою чергу, елемент *Placemark* може містити треки, збережені елементом *gx:Track*. Він може містити будь-яку кількість елементів *gx:coor*, які насправді є точками, визначеними *GPS*-координатами. Час відвідування також можна призначити кожній точці за допомогою елемента *when*.

Відмінністю збереження результатів у порівнянні з форматом *GPX* є можливість налаштувати колір і ширину траєкторій, що відображаються. Таким

чином, кожна людина генерується та призначається випадковим кольором, щоб її було легко розрізнити.

Оскільки досліджувана методика обробки даних передбачає прямий експорт отриманих траєкторій у файл, який потім відображається в *Google Earth*, відсутня необхідність створювати графічний інтерфейс. Натомість інтерфейс користувача складається з файлу конфігурації *config.py* та файлу запуску *run.py*.

### 3 ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ РОЗРОБЛЮВАНОГО МЕТОДУ

#### 3.1 Опис експериментів для перевірки працездатності та ефективності методу

Тестовий набір даних, для задач поточної роботи, сформовано на основі загальнодоступного набору даних *COCO1* [32]. Останній виконано за допомогою камери з роздільною здатністю  $1920 \times 1080$  пікселів, 30 кадрів на секунду та горизонтальним полем зору  $81,9^\circ$ . Він містить ряд відеозаписів тривалістю від одиниць до десятків секунд і включає зйомку, в основному людей, на відстані 10 - 50 м від об'єктива камери. Відповідна схема показана на рис. 3.1. Разом зі згаданими даними тут є також запис польоту, що містить параметри, зафіксовані датчиками дрона, яких достатньо для успішного визначення місцезнаходження людей за допомогою відповідних методів. Однак, початок журналу і реальний запис можуть бути не точно синхронізовані, що необхідно враховувати при роботі з набором даних.

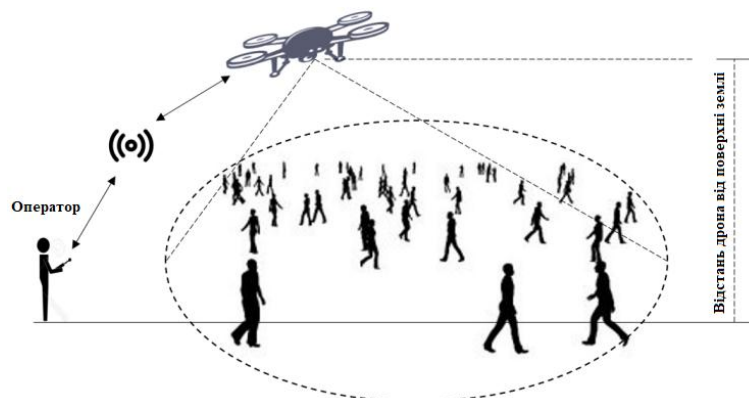


Рис. 3.1. Схема огляду області відстежування

#### *Набір ідентифікаторів тестового набору даних*

Для цілей оцінки інструментів повторної ідентифікації в рамках поточної роботи було сформовано набір елементів тестового набору даних. На 100

зображеннях, випадково вибраних із усього набору даних, люди були виявлені детектором «*Faster R-CNN with NASNet-A*». Потім їхні попередні перегляди сортувалися вручну, а помилкові або неповні виявлення видалялися.

У результаті тестовий набір даних складається з 6 різних ідентифікацій, які складаються із 384 зображень із розмірами від 19x50 до 115x241 пікселів. Об'єкти записуються в різних позах, на різних відстанях і під різними кутами. Наочний приклад можна побачити на рис.3.2.



Рис. 3.2. Приклад зображення натовпу із тестового набору даних [32]

Додаток було протестовано під кількома операційними системами з різним обладнанням:

1. ПК з *Windows 10 (64bit)*, процесор *i5-2500K* (4 ядра, тактова частота 4 ГГц), 16 ГБ оперативної пам'яті, відеокарта *NVIDIA RTX 2060* (6 ГБ графічної пам'яті).

2. ПК з операційною системою *High Sierra (64bit)*, процесор *i7-7700HQ* (4 ядра, тактова частота 3,8 ГГц), 16 ГБ оперативної пам'яті, відеокарта *AMD Radeon Pro 555* (2 ГБ ОЗУ).

3. Віртуальний комп'ютер з *Ubuntu (64bit)*, віртуалізований процесор (2 ядра, тактова частота 3 ГГц), 8 ГБ оперативної пам'яті, віртуалізована відеокарта (128 МБ графічної пам'яті).

Порівняння приблизних профілів висоти секції «Захід-Схід» (*W-E*) відповідно до схеми на рис.3.1 показано на рис. 3.3 – 3.5.

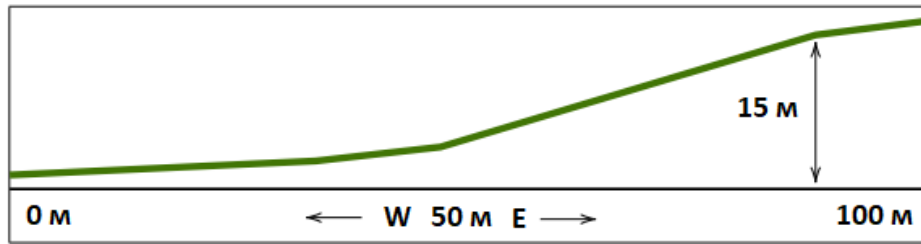


Рис. 3.3. Фактичний профіль висоти (відповідно до *Google Earth*)

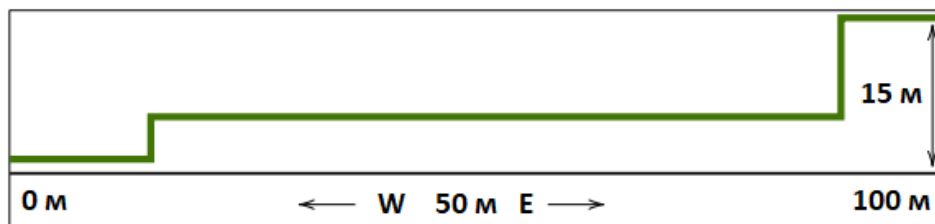


Рис. 3.4. Профіль рельєфу за цифровою базою даних  
«*The TanDEM-X 90m Digital Elevation Model*»

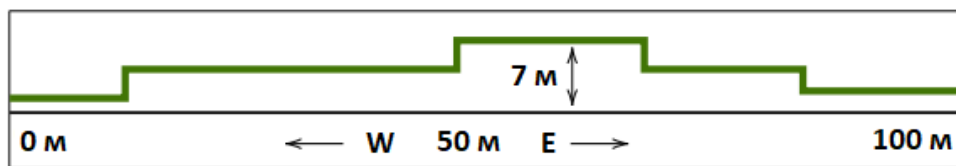


Рис. 3.5. Профіль рельєфу за цифровою базою даних  
«*Copernicus Land Monitoring Service–EU-DEM*»

На попередніх зображеннях профіль поверхні позначено зеленим кольором. Напрямок на захід (*W*) і схід (*E*), приблизна горизонтальна відстань і висота важливих точок вказується чорним кольором.

Базовий додаток працює на всіх перерахованих платформах за умови наявності необхідних передумов (інтерпретатор *Python* тощо). Для вимірювання складності операцій і їх порівняння в переважній більшості випадків використовувалася система №1, оскільки вона має в своєму розпорядженні досить продуктивну відеокарту *NVIDIA*, яка підтримується більшістю інструментів машинного навчання, що пропонують прискорення за допомогою графічного ядра.

### 3.2 Особливості проведення ексериментів

Було випробувано кілька варіантів модифікації зображення з метою покращити виявлення людей шляхом усунення шуму або зміни гистограми. Результати оцінювалися на основі успішності детекторів осіб на змінених таким чином зображеннях, зокрема з використанням рівня достовірності, який нейронні мережі вказують для окремих знайдених об'єктів. Результати модифікованих входів порівнювалися з оригіналом для різних версій використовуваних детекторів. Кількість відсутніх відносно отриманих дійсних об'єктів, кількість покращених відносно знижених значень вірогідності виявлення та його загального збільшення. У табл.3.1 з конкретними значеннями, наведено відповідні результати.

При використанні методів усунення шумів суттєвих поліпшень не спостерігалось, скоріше навпаки, для всіх типів використовуваних інструментів. Хоча рівень достовірності виявлення не змінювався в більшості спостережуваних випадків, порівняно часто він знижувався, тобто погіршувався.

Таблиця 3.1

Приклад змін у виявленні людини після виконання коригування зображення

Метод	Параметри	Пропущено	Додано	Вище (сумарно)	Нижче (сумарно)
<i>Gauss Blur</i>	<i>ksize 5x5</i>	8	1	21 (1,60)	56 (-3,02)
<i>Bilateral Filter</i>	<i>sigmaColor 30, sigmaSpace 30</i>	6	1	27 (1,12)	49 (-1,68)
<i>CLAHE</i>	<i>clipLimit 1, tileGridSize 7x7</i>	1	1	33 (1,57)	40 (-0,99)

Тестовий набір складається з 50 випадково вибраних зображень. Наведені дані отримані за допомогою інструменту *Faster R-CNN with NASNet-A*, який ідентифікував 138 осіб на невідредагованих кадрах. Пояснення параметрів методу можна знайти у дослідженнях бібліотеки *OpenCV* [21]. Стовпці «пропущено» / «додано» показують кількість пропущених або нещодавно доданих дійсних виявлень порівняно з оригінальним зображенням. Стовпці «вище» / «нижче»

показують, скільки об'єктів було покращено, відповідно зміна вірогідності виявлення та загальна сума всіх цих відмінностей (нормальне значення одного результату коливається від 0 до 1).

Запис польоту синхронізується з відеозаписом за даними про включення відеокамери. Перша проблема виникає, якщо активація камери не точно зафіксована в момент фактичного початку зйомки. У тестовому наборі даних помічено, що для кількох записів початок відеозапису з першими записаними даними може відрізнятись від 100 до 300 мс. Однак навіть такий короткий момент може призвести до відхилень визначених позицій - наприклад, навіть невелика різниця в обертанні дрона протягом цього часу може в результаті істотно відхилити позиції людей, особливо при зйомці людей у натовпі. Цю проблему можна вирішити, встановивши вручну затримку/черговість запису, що, однак, вимагає більш ретельного аналізу з боку користувача. Можна було б вирішити це автоматично, виявивши рух усіх людей в одному напрямку, і компенсувати такі поверхневі переміщення, змінивши рух на зустрічний. Однак, для кожної людини згадана помилка проявляється по-різному - якщо, наприклад, особа рухається в протилежному напрямку відхилення, її зміщення буде меншим і компенсація буде занадто великою для неї або недостатньою для інших осіб. На практиці таке коригування може означати компенсацію навіть у моменти, коли всі люди рухаються в одному напрямку шляхом природної ходьби (і коли журнал польотів даватиме правильні дані). Тому компенсація не є обов'язковою.

Потенційним джерелом помилок також є можлива неточність польотних датчиків. Таким чином, до даних із журналу польотів застосовується згладжування за допомогою фільтра Калмана (рис. 3.6).

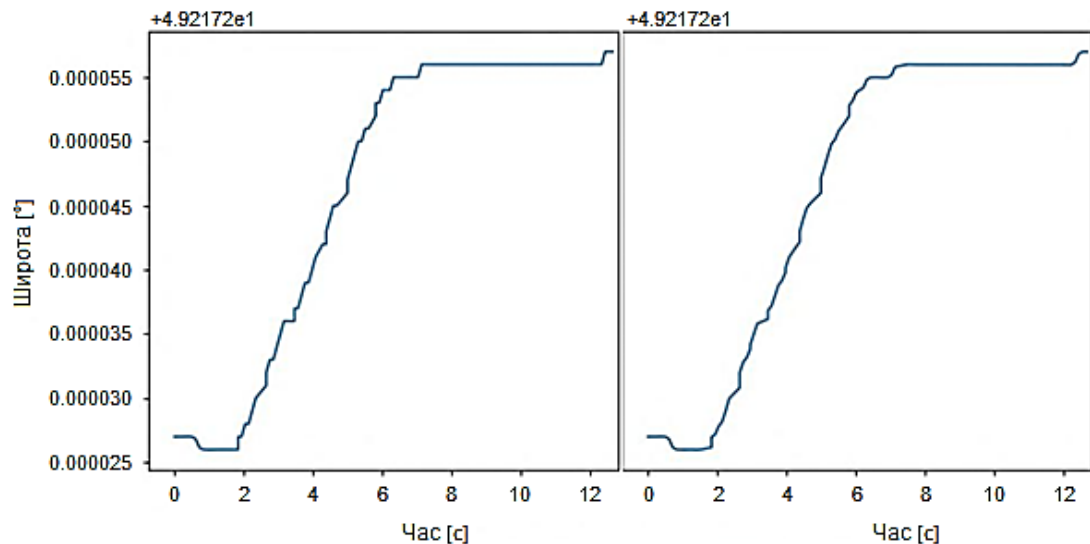


Рис. 3.6. Застосування фільтра Калмана до широти *GPS*-положення дрона з тестового набору

За рис. 3.6, ліворуч видно вихідні дані з журналу польотів, праворуч - після редагування. На осі  $y$  значення приросту широти вказано в десяткових градусах до  $49,2172^\circ$ , вісь  $x$  вказує час. Так, можна спостерігати невелике згладжування кривої та більш плавний хід ступінчастих змін.

Всього було протестовано 4 моделі нейронної мережі для виявлення людини. Спочатку було протестовано інструмент *Open Pose*, що, як правило, вважається досить надійним рішенням для виявлення людей. Однак він має деякі ключові обмеження. Тому згодом було реалізовано модуль для використання мереж, доступний з офіційної пропозиції модельного ряду *TensorFlow*, який усуває ці помилки.

Моделі не тільки вибрано на основі метрики *mAP* (з їхніх навчальних наборів даних), але їх було перевірено, щоб побачити, як вони поведуться в тестовому наборі і чи придатні їхні результати для подальшого використання в повторній ідентифікації особи.

#### *Виявлення за допомогою Open Pose*

*Open Pose* успішно розпізнає людей з висоти та при огляді під різними кутами. Однак перша проблема виникає з більш віддаленими особами, коли мережа краще працює на зображеннях із вищою роздільною здатністю. Тут її



обчислювальна складність різко зростає. На тестовій серії час виявлення одного кадру за допомогою процесора варіювався, при роздільній здатності  $1280 \times 720$  пікселів, у порядку хвилин. При використанні відеокарти процес прискорювався до одиниць секунд, але через обмеження графічної пам'яті він не дозволяв отримати роздільну здатність вище *640p*. З цієї причини були менш точні виявлення віддалених осіб. Крім того, згадані значення були досягнуті лише за допомогою попередньо скомпільованої бінарної версії інструменту, яка обмежена платформою. Базова реалізація в *Python* досягла навіть помітно гірших значень.

Інша проблема виникає, коли, наприклад, кінцівка розпізнається по-різному на двох близько послідовних зображеннях – результуюча обмежувальна рамка для тієї самої людини може значно відрізнятись. Це створює значні проблеми при повторній ідентифікації осіб.

Цій проблемі можна частково запобігти шляхом визначення основних частин тіла, необхідних для успішного виявлення (наприклад, вимагаючи принаймні тулуба, голови та принаймні однієї нижньої кінцівки). Однак це рішення є лише частковим, оскільки кожне додане обмеження може призводити до зменшення виявлень - наприклад, вимога виявлення обох стегон і плечей стабілізує результат, але виключає всіх людей, які розміщені боком по відношенню до камери, навіть якщо з іншого ракурсу їх добре видно.

Через ключові недоліки *Open Pose* реалізована можливість використання нейронних мереж із модельного ряду *TensorFlow*. Вони не мають проблем із зміною розмірів обмежувальної рамки та можуть досягати великих швидкостей.

Всього було протестовано 3 варіанти моделі, які перевірялися на наборі даних *COCO*:

1. *SSD with Resnet 50 v1 FPN feature extractor.*
2. *Faster R-CNN with Inception v2.*
3. *Faster R-CNN with NASNet-A featurization.*

Модель №1 досягає рейтингу *mAP* 35 на наборі даних *COCO*. Штучні нейронні мережі типу *SSD5* належать до найшвидших, але точних архітектур. Однак на тестовому наборі даних було досягнуто відносно низької кількості

правильних виявлень порівняно з *Open Pose*. Крім того, мережа погано виявляла більш віддалені об'єкти, тому вона виявляється непридатною для цього набору даних і більше не використовувалася.

*Модель №2* досягає рейтингу *mAP* 30 на наборі даних *COCO*, а її швидкість також досить висока. На тестовому обладнанні обробка одного зображення з роздільною здатністю  $1920 \times 1080$  пікселів займає приблизно 100 мс за допомогою відеокарти, або у 5-10 разів довше з процесором. Вона досягає хороших результатів у кількості успішних виявлень на тестовому наборі даних. Однак проблема виникає з великою кількістю хибнопозитивних виявлень.

Усі згадані моделі вказують рівень достовірності виявлення для кожного об'єкта в діапазоні  $(0;1>)$ , за допомогою якого можна відфільтрувати ці виявлення. Однак за складніших умов мережа ігнорує велику кількість людей, а за більш сприятливих умов вона часто генерує кілька повторюваних виявлень для однієї особи. Тому простого встановлення запасу достовірності недостатньо для досягнення оптимальних результатів. Ця проблема вирішується шляхом додавання обмеження на максимальне перекриття виявлених осіб. Можна встановити більший допуск достовірності виявлення, що призводить до знаходження більшої кількості об'єктів (включаючи дублікати), які згодом відфільтровуються. Тут з кожної групи взаємних перекриттів лише 1 результат відповідно до найвищого рівня визначеності завжди залишається.

На рис. 3.7 наведено результати, що були отримані за допомогою моделі «*Faster R-CNN with Inception v2*». Цифри вказують на достовірність виявлення відповідного об'єкта. У верхній частині мінімальна вірогідність 0,7 та без заборони перекриття. Посередині потрібна вірогідність 0,4 та без заборони перекриття. У нижній частині значення вірогідності 0,3 із заборonoю перекриття вище 50%. Чорні поля вказують на всі виявлення, а червоні - лише на ті, які досягають необхідної достовірності виявлення.

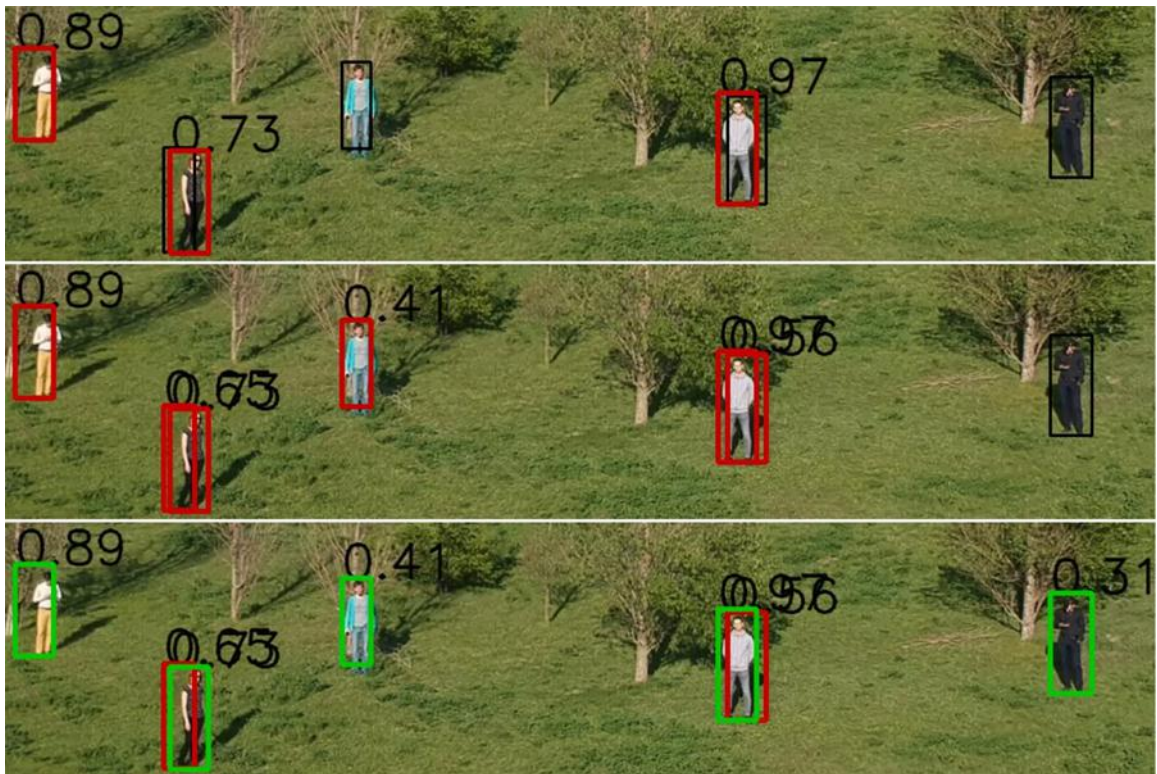


Рис. 3.7. Приклад можливих проблем у виявленні залежно від значення вірогідності

За рис. 3.7, особи, які досягають необхідної достовірності та водночас мають найвище значення з усіх локальних перекриттів, відзначаються зеленим кольором. Використовуючи межу достовірності 0,7, на верхньому зображенні знайдено 3 дійсних особи (решта не досягає необхідного значення). Щоб наступну особу вважати дійсною, обмеження потрібно зменшити до 0,4 у середньому кадрі. Однак це також додало 2 небажаних дублікати. Рішення можна побачити на нижньому зображенні, де використовувався мінімальний поріг достовірності 0,3 і водночас фільтр для видалення обмежувальних рамок, що перекриваються. Це видалило дублікати, а використання нижньої межі також дозволило виявити іншу особу.

Значення дозволеного перекриття може видалити навіть дійсних людей у момент, коли вони проходять повз один одного - але це нормально. У випадках більшого збігу ідентифікатори можуть бути сплутані (наприклад, із збігом у 50 %, уже неможливо визначити, хто ким є). Таким чином, накладний фільтр запобігає обом цим проблемам одночасно.

Модель №3 досягає рейтингу *mAP* 43 за набором даних *COCO*, тобто найкращого результату з перерахованих моделей. Однак це компенсується в кілька разів більшою обчислювальною складністю. На тестовому апаратному забезпеченні обробка одного зображення з роздільною здатністю  $1920 \times 1080$  пікселів займає приблизно 1-1,5 с при використанні відеокарти, або у 20 разів довше з процесором. Однак на тестових даних він дає найкращі результати з усіх перерахованих детекторів у переважній більшості випадків. На основі цих даних згодом можна вибрати мінімально необхідний рівень вірогідності.

Так, порівняння проводилося на 120 випадково вибраних зображеннях із тестового набору даних. Фільтр було застосовано для видалення накладених виявлень. Детектор «*Faster R-CNN with Inception v2*» виявив загалом 392 дійсних особи. Модель «*Faster R-CNN with NASNet-A*» надала 413 дійсних результатів. Достовірність виявлення моделі №3 здебільшого близька до 1 і майже не дає повторних результатів. Іноді вона генерує хибнопозитивні результати для небажаних об'єктів, але у неї немає ускладнень, наприклад, із правильним ігноруванням тіней людей. Таким чином, вона є найефективнішим варіантом з усіх перевірених інструментів.

Для визначення *GPS*-положення відстежуваних осіб було протестовано 2 процедури: визначення відстані предмета на плоскій поверхні і визначення положення об'єкта за допомогою висоти. Перша з них дала абсолютно незадовільні результати на більшості тестових даних. Друга усуває ці недоліки, але вимагає бази даних про висоту поверхні в даному місці, що передбачає додаткові ускладнення. Цей метод не потребує зовнішніх залежностей і є досить швидким, але його можна використовувати, лише якщо особи, які контролюються, рухаються по рівній поверхні. Тому в тестовому наборі даних він працював прийнятно лише у виняткових випадках, коли дрон рухається разом з об'єктами спостереження на рівній поверхні. У той же час, коли люди рухаються узбіччям, а дрон слідує за ними знизу, може виникати значна неточність, яка практично повністю скасовує достовірність результатів виявлення. Схематично, це пояснюється за рис. 3.8.

Ця проблема призводить до недостатніх результатів на більшій частині тестового набору даних, тому метод було визнано невідповідним.

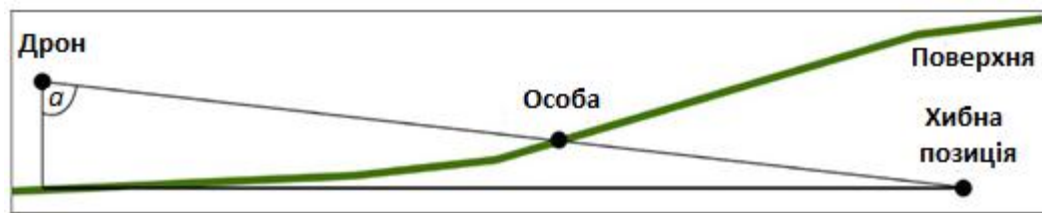


Рис. 3.8. Приклад неправильного визначення *GPS*-положення людини, коли актуальне значення висоти є невідомим

Система фіксує, що людина знаходиться на тій самій висоті, що й поверхня. Тому положення осіб вищій точці визначаються значно далі від дрона, ніж його фактичне розташування. Чим більше кут « $a$ » для спостережуваного об'єкта, тим далі він буде визначений.

Для повторної ідентифікації людей було випробувано кілька підходів. Спочатку вони будуть описані в загальних рисах з точки зору використання, а потім будуть представлені методи їх оцінки разом з досягнутими результатами. Для виділення ознак використовувалися 2 основні принципи.

Для кожного підходу наведено приблизний час обчислення вилучення на тестовому апаратному забезпеченні для зображень людини розміром 128x256 пікселів (приблизно відповідає верхній межі розміру людини в тестовому наборі даних):

1. Обчислення гістограм – це базова техніка аналізу зображень. Характеристики обчислюються менш ніж за 1 мс (обчислення відбувається на процесорі). Метод більше підходить для менш мінливого середовища і на нього сильно впливає багато факторів (освітлення, фон, кут огляду тощо).

2. *AlignedReID* – підхід використовує нейронну мережу. Чим більше деталей про людину доступно, тим краще працює алгоритм (на відміну від використання гістограм, які від цього не настільки змінюють продуктивність). Отримання функцій людини за допомогою центрального процесора займає близько 0,5 с, тоді

як за допомогою відеокarti це значення зменшується до 30 мс. Тому це більш вимогливий метод з обчислювальної точки зору, але він досягає сучасних результатів у повторній ідентифікації осіб.

Модулі *HistogramTracker* і *AlignedReIdTracker* (або інші їх варіанти) побудовані на основі описаних принципів, які дозволяють порівнювати отримані дані за допомогою різних підходів, з різними параметрами допуску тощо. Таким чином, можна, наприклад, використовувати контекстно-вільний підхід як порівняння вилучених функцій або, навпаки, контекстно порівнювати положення людей на зображенні, позицію *GPS*, час зображення, кількість людей тощо.

У поточній роботі акцент виконується на оцінці унікальності траєкторій осіб, тому запропоновано декілька методів оцінки результатів повторної ідентифікації осіб.

На рис. 3.9 дрон позначено, як *D*. Позначення *Q1*, *Q2*, *Q3* відповідають реальному положенню людей. А *P1*, *P2*, *P3* є визначеними системою локаціями для цих осіб. Фактичний профіль висоти зображено зеленим кольором, дані висоти (*TDM90*) – червоним. Тут особа 1 відображається з очікуваним результатом. Однак особи 2 і 3 проєктуються в точку з однаковими горизонтальними координатами (різними лише по вертикалі). Те саме стосується всіх інших можливих осіб у синій зоні між особами 2 і 3.

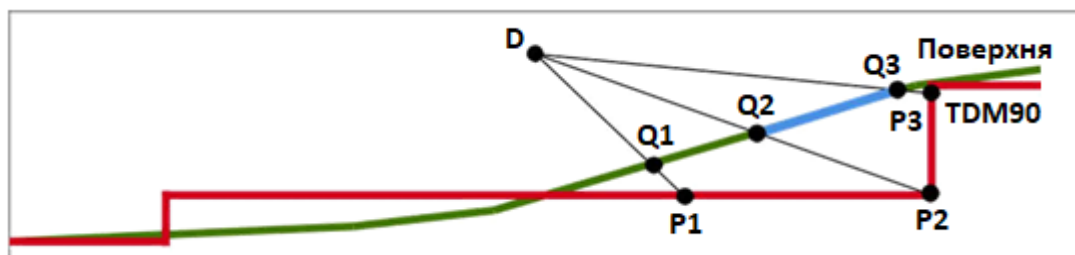


Рис. 3.9. Варіативність виявлення осіб, залежно від перепаду висот поверхні

Щоб максимально оцінити інструменти для повторної ідентифікації щодо відповідності використання в робочому контексті, не використовувалися зовнішні набори ідентифікаційних даних. Натомість було сформовано та вручну змінено

набір тестових даних, який складається виключно з осіб, виявлених на тестових відео. Лише контекстно-вільні варіанти алгоритмів можна оцінювати за допомогою тестового набору даних. Вони не надають інших даних (час або місцезнаходження людини тощо) на додаток до попереднього перегляду осіб. Однак результати можна використовувати для порівняння якості використовуваних принципів і для налаштування параметрів модулів повторної ідентифікації. Зразок і отримані ідентифікатори (особи) складаються з набору кадрів (або попередніх переглядів осіб).

Для оцінки коректності результатів було використано кілька показників, які доповнюють один одного. Їхня характеристика наведена нижче (табл. 3.2).

Покриття ідентифікаційних даних зразка (англ. *PVI*) – кожному ідентифікаційному номеру зразка (відсортованого за порядком спадання щодо кількості кадрів) призначається унікальне значення. На його основі отримана особа має найвище представлення маркерів у зразку ідентифікації. Маркери цієї особи в ідентифікаторі шаблону лише тоді вважаються правильними, коли інші вважаються неправильними. Потім на основі маркерів правильної отриманої ідентифікації визначається відсоток охоплення вибіркової особи, тобто охоплення всього набору даних вибірки (ідеальне значення – 100%).

Покриття результатів (англ. *PV*) – схоже на *PVI*, але значення охоплення виводиться в протилежному напрямку. Таким чином, це вказує на те, наскільки отримані ідентичності охоплені особами вибірки (ідеальне значення – 100%).

Таблиця 3.2

Результати роботи алгоритмів повторної ідентифікації на тестовому наборі даних

Основа методу	Глибоке порівняння	Редагування зображення	<i>PVI</i> (%)	<i>PV</i> (%)	<i>PPO</i>
Гістограма	-	-	31,2	30,8	14,03
Гістограма	+	<i>Gauss Filter</i>	57,3	58,5	4,63
Гістограма	+	-	58,5	58,9	3,6
Гістограма	+	<i>Bilateral Filter</i>	61,7	63,0	4,07
<i>AlignedReID</i>	-	-	62,2	62,2	2,75
<i>AlignedReID</i>	+	<i>CLAHE</i>	87,9	86,5	1,22
<i>AlignedReID</i>	+	<i>Gauss Filter</i>	88,9	88,9	1,13
<i>AlignedReID</i>	+	<i>Bilateral Filter</i>	92,6	92,6	1,12
<i>AlignedReID</i>	+	-	95,2	95,2	1,07

Коефіцієнт кількості осіб (англ. *PPO*) – вказує на співвідношення кількості результуючих ідентичностей до вибіркового. Тут можна створити багато нових ідентичностей з невеликою кількістю маркерів, або, навпаки, кілька вибіркового ідентичностей, об'єднаних в один результат. Крім того, це дозволяє відрізнити якість алгоритмів, для яких інші параметри досягають ідентичних значень (ідеальне значення – 1).

З кожним вимірюванням набір маркерів сортується випадковим чином, що може помітно вплинути на подальшу повторну ідентифікацію (зазвичай у відсотках). Тому кожне вимірювання проводиться повторно і розраховується відповідне середнє значення. Так, кожне вимірювання проводилось 10 разів і результати усереднювались (див. табл. 3.2). Значення *PV* і *PVI* часто збігаються або є досить близькими. Якщо отримані ідентифікатори міститимуть, наприклад, 10 правильних маркерів із вибірки ідентичностей (із 50 маркерів), значення *PV* досягне 100%, тоді як значення обчислень *PVI* правильно вказуватиме на помилку у 20% випадків.

Серед іншого, перевірено кілька варіантів чутливості модулів повторної ідентифікації, значення в табл. 3.2 наведені для знайдених приблизно оптимальних конфігурацій. Подібним чином протестовано різні інтенсивності коригування зображення. Так фільтр Гаусса вплинув на результати або незначним чином, або негативно. Інакше зарекомендував себе алгоритм *CLAHE*, який не досяг кращих значень у жодному з тестованих випадків, ніж з оригінальним зображенням. Однак алгоритм *Bilateral Filter* виявився можливим вдосконаленням методів на основі гістограм. Однак, завдяки змішуванню кольорів і видаленню шуму гістограми стають більш схожими, тому доцільно дещо зменшити допуск порівняння під час його використання, щоб уникнути занадто великої кількості помилкових збігів.

З виміряних значень стає зрозуміло, що використання вичерпного порівняння (новий маркер порівнюється з усіма маркерами існуючих ідентичностей) суттєво покращує результати. А як у випадку повторної ідентифікації на реальному записі, порівняно з використанням лише 1 останнього кадру кожної ідентифікації, така помітна різниця не виявляється. Це пов'язано з тим, що у тестовому наборі даних



кадри обробляються у випадковому порядку, тому навіть два послідовних можуть істотно відрізнятись. І навпаки, у реальному відеозаписі людину зазвичай фіксують у такому ж положенні, як на попередньому кадрі.

У табл. 3.2 помітно, що використання гістограм не дає дуже хороших результатів (найкращий варіант з коригуванням зображення можна порівняти з найгіршим результатом *AlignedReID*). Тому гістограми слід використовувати лише в поєднанні з іншою інформацією (наприклад, положенням людини) або для відеозаписів із значно віддаленими особами, де недостатньо деталей для інших підходів. Так, може виникнути ситуація, коли, на перший погляд здається, що гістограми зображень 1 і 2 більш схожі, ніж гістограми 2 і 3, навіть якщо зображення 1 належить іншій людині. Те ж саме підтверджується, наприклад, розрахунком перетину гістограм 1-2 зі порівняно із зображеннями 2-3. На гістограму впливають кольори фону, а також сама людина. І навпаки, метод *AlignedReID* правильно призначає зображенням 1 і 2 значення несхожості, тоді як зображення 2 і 3 оцінюють несхожість з меншим значенням. Загалом, результати роботи цієї нейронної мережі є досить хорошими. Однак слід зазначити, що неповні та недійсні зображення були видалені із тестового набору даних. Тому результати на ньому можуть дещо відрізнятись, порівняно з іншими наборами даних. Крім того, може виникнути плутанина ідентичності, наприклад, через одне зображення з поганим відображенням у кадрі, що призведе до недійсності таких векторів траєкторії. Тому необхідно врахувати ці альтернативи та адаптувати отримані умови до реального використання (ймовірно, шляхом зменшення допуску на подібність).

Отже, окремим попереднім переглядам призначаються їхні гістограми (варіації компонентів кольору, об'єднані в 1 вимір, нормалізовані) – точки на осі абсцисс представляють варіанти комбінацій кольорів, вісь у вказує на їхню відносну кількість.

Найбільша проблема, яка спостерігалася в цій мережі, полягає в розрізненні зображень різних пропорцій, оскільки модель використовує зображення людей певного розміру. Тому переважну більшість зображень потрібно збільшувати або

зменшувати. Зазвичай це не викликає проблем. Однак у випадках, коли ту саму особу виявляють, наприклад, у значно ширшому середовищі, ніж на іншому зображенні, після зміни розміру співвідношення сторін людини дуже відрізняється, і вона виглядає як інша особа для *AlignedReID*. Це часто спричинено, наприклад, виявленням за допомогою моделі *Open Pose*.

### 3.3 Аналіз результатів обробки даних

Однією із задач поточної роботи є відображення отриманих траєкторій руху людей у картографічній програмі. Так, доцільним варіантом є *Google Earth*. Оскільки основною частиною загального алгоритму є детектор людини, обчислення *GPS*-координат і подальша повторна ідентифікація людей, збереження та відображення результатів може здатися не дуже складною операцією. Однак тут є кілька ускладнень, які розглядаються нижче.

Для відображення результатів у *Google Earth* спочатку було реалізовано опцію експорту маршрутів у файл *GPX*. Відповідний результат можна побачити на рис. 3.10. Цей формат можна відкрити та відобразити вручну у програмі, але з обмеженими можливостями. Всі особи («*Person*») мають траєкторію одного кольору, що виявилось абсолютно неприйнятним при відображенні кількох траєкторій в одному місці.

Хоча *GPX* є загальноживаним рішенням для роботи з даними *GPS*, у *Google Earth* основним носієм є файли *KML/KMZ*, які також дозволяють передавати графіку та інші деталі. Тому також було реалізовано опцію експорту у файл *KML*, за допомогою якої випадково згенеровані кольори призначаються окремим особам, щоб їх можна було легко відрізнити один від одного. Такий результат також можна автоматично відкрити в *Google Earth* безпосередньо з додатку, а конкретні маркування траєкторії також можна налаштувати за необхідністю (ширину траєкторії тощо). Приклад остаточного відображення траєкторій можна побачити на рис. 3.11.



Рис. 3.10. Візуалізація траєкторій на *Google Earth* у форматі *GPX*



Рис. 3.11. Візуалізація траєкторій на *Google Earth* у форматі *KML*

Якщо запис має високу частоту кадрів і людину розпізнають практично в кожному кадрі, результуюча траєкторія містить велику кількість точок. Вони коливаються навколо реального маршруту через різні неточності (сенсори дронів, незначні зміни у виявленні, недосконала модель висоти тощо). Тому проблемою може бути висока фрагментація траєкторій, коли результат створює враження хаотизму. Відповідний приклад можна побачити на рис. 3.12.



Рис. 3.12. Необроблена траєкторія особи (56 метрів за 10 секунд)

Для оптимізації такого маршруту можна використовувати два методи. Перший - агрегація треків за часовим інтервалом. Це дозволяє усереднювати кількість надто детальних позицій *GPS* в одній точці за певний проміжок часу. Оскільки в реальному світі людина може рухатися лише з обмеженою швидкістю, невеликі відмінності в положенні незначні, і результат такої операції не повинен знижувати правильність (якщо інтервал вибрано відповідним чином).

Другий варіант - застосувати фільтр Калмана щодо отриманих позицій. Однак, оскільки маркери людини не повинні пролягати через рівні проміжки часу,

результуюча траєкторія може виглядати ступінчасто, якщо нове положення не було відоме протягом тривалого часу, а оцінка базуватиметься на застарілій інформації. Тому ефективним варіантом є поєднання фільтра Калмана з попереднім агрегуванням позицій за часовий інтервал. Відповідний результат можна побачити на рис. 3.13.



Рис. 3.13. Результуюча траєкторія особи (агрегована за інтервали 0,5 с і згладжена за допомогою фільтра Калмана)

Проблема появи небажаних об'єктів виникає вже при виявленні людей. Це хибнопозитивні результати, тобто небажані об'єкти, позначені як дійсні особи. Зазвичай це лише одноразові відхилення (наприклад, випадкова помилка, спричинена шумом) або короткочасна (наприклад, тінь людини в даній точці зору), рідше постійна помилка виникає протягом тривалого періоду часу (наприклад, позначення залишеного одягу як людини).

У контексті виявлення ці проблеми можна вирішити лише з труднощами (наприклад, шляхом повної заміни використаного алгоритму). Однак короткочасні відхилення можна відносно ефективно вирішити в рамках коригування отриманих траєкторій. Тому була реалізована можливість видалення наступних об'єктів:

- не містить достатньо маркерів;
- спостерігається лише у короткий проміжок часу.

Таким чином, можна налаштувати обмеження для мінімальної кількості маркерів, а також для найкоротшого необхідного інтервалу часу. При правильно підібраних параметрах ймовірність видалення важливої інформації дуже низька. Наприклад, для людей, виявлених менше ніж за 1 секунду, це, ймовірно, буде помилкою, але навіть якщо це буде дійсний висновок, такий варіант є прийнятним.

І навпаки, позитивний внесок від усунення помилок є помітним, особливо для детекторів з низькою прогностичною цінністю позитивного результату.

Підсумовуючи, аналіз результатів обробки даних вказує, що використання гістограм для повторної ідентифікації та пов'язаних з цим ускладнень є ефективним з точки зору розрізнення ідентичності та чіткості траєкторій людей. Загальний алгоритм успішно відображає рух дрона, включаючи зміни висоти дрона та орієнтації камери. Для 4 із 5 осіб місцезнаходження *GPS* було визначено правильно. Максимальне відхилення становить 5 м. Найбільшою складністю тут виявилася дещо неточна база даних про висоту (*TDM90*). Але це викликано якістю вхідної інформації, а не особливостями алгоритму обробки та інтерпретації даних.

## ВИСНОВКИ

В ході реалізації даної дипломної роботи було виконано широкий огляд існуючих методів комп'ютерного зору, зорієнтованих на відстеження об'єктів. Перший етап дослідження передбачав зіставлення та аналіз різноманітних підходів, що використовуються в сучасних системах відстеження. Детально проаналізовано існуючі методи відстеження траєкторії. Отримані дані визначили ключові технічні особливості та переваги кожного методу, які були враховані при подальшій розробці власного методу відстеження траєкторії людини в натовпі.

Були детально описані вибрані алгоритми та технології, що застосовуються для відстеження траєкторії людини. Цей етап передбачав розгляд різних аспектів, таких як точність визначення руху, чутливість до зовнішніх факторів та можливості адаптації до змінних умов. Детальний опис алгоритмів дав змогу визначити та врахувати ключові аспекти для подальшого удосконалення розробленого методу.

Протягом написання було визначено основні переваги та недоліки аналізованих методів. Для цього було проведено зіставлення отриманих результатів, виділено позитивні та негативні аспекти кожного методу. Такий підхід дозволив сформулювати об'єктивну оцінку і врахувати важливі аспекти при розробці власного методу відстеження.

Випробувано 4 різних варіанти детекторів, та проведено аналіз ефективності їх застосування для різних ситуацій. Для отриманих експериментів було обрано раціональний варіант, який на тестовому наборі даних досягає середньої точності. На основі отриманих даних впроваджено необхідні вдосконалення, наприклад, для видалення дублікатів виявлення.

Для оцінки повторної ідентифікації осіб сформовано набір ідентифікаторів, що містить 384 зображення та 5 відповідних осіб. Де обробка даних виконувалась не лише базовими методами повторної ідентифікації за допомогою гістограм, а й за допомогою машинного навчання, а саме нейронної мережі *AlignedReID*, яка дає показник успішності 95,2% на тестовому наборі даних навіть без використання

додаткових коригувань зображення. І навпаки, підхід, заснований на гістограмі, досяг максимуму приблизно в 60% навіть з коригуванням.

Створено математичну модель обробки даних. На основі аналізу отриманих результатів та характеристик вибраних методів була розроблена модель, яка враховує ключові параметри для точного та ефективного відстеження траєкторії, та було проведено експерименти, які включали в себе тестування методу в різних умовах, а також порівняння результатів з іншими методами відстеження. Слід зазначити, що на тестовому наборі даних відстань між особами та камерою здебільшого знаходиться в діапазоні 10-60 м. А верхня межа похибки визначених позицій *GPS* досягає порядку метрів у більшості протестованих випадків.

Визначено, що найбільш проблематичною частиною процесу обробки інформації є бази даних висот, які через свою відносно низьку точність можуть вносити похибку в десятки метрів навіть у результуючі траєкторії.

Узагальнюючи, виявлення людей на зображенні завдяки машинному навчанню дає хороші результати, але хибнонегативні та дійснопозитивні результати трапляються досить часто.

Отже, можна зазначити, що у ході виконання дослідження опрацьовано всі раніше сформовані задачі роботи.

## ПЕРЕЛІК ПОСИЛАНЬ

1. ШУШУРА, О & ПРИСЯЖНЮК, В. (2023). ІНФОРМАЦІЙНА СИСТЕМА КЛАСИФІКАЦІЇ ТА МАРКУВАННЯ ЗОБРАЖЕНЬ ДЛЯ НАВЧАННЯ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ. Вісник Херсонського національного технічного університету. 137-142. 10.35546/kntu2078-4481.2023.3.17.
2. Zavalnyi, Mykhailo. (2021). Відеоспостереження в публічних місцях як елемент державно-приватного партнерства у сфері забезпечення охорони правопорядку.
3. Irfan, Syed & Islam, MD & Mia, Md & Islam, Sahidul & Islam, Tanjin. (2023). ADVANCEMENTS IN OBJECT DETECTION AND TRACKING ALGORITHMS: AN OVERVIEW OF RECENT PROGRESS. EPRA International Journal of Research & Development (IJRD). 134-140. 10.36713/epra14418.
4. Wang, Qiufu & Zhou, Jiexin & Li, Zhang & Sun, Xiaoliang & Yu, Qifeng. (2023). Robust monocular object pose tracking for large pose shift using 2D tracking. Visual Intelligence. 1. 10.1007/s44267-023-00023-w.
5. Gkelios, Socratis & Features, Deep & Plakias, Spiros & Boutalis, Yiannis & Chatzichristofis, Savvas. (2021). Deep Convolutional Features for Image Retrieval. Expert Systems with Applications. 177. 114940. 10.1016/j.eswa.2021.114940.
6. Qi, Yuankai & Zhang, Shengping & Qin, Lei & Yao, Hongxun & Lim, Jongwoo & Yang, Ming-Hsuan. (2018). Hedging Deep Features for Visual Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 1-1. 10.1109/TPAMI.2018.2828817.
7. Coppi, Dalia & Calderara, Simone & Cucchiara, Rita. (2015). Transductive People Tracking in Unconstrained Surveillance. IEEE Transactions on Circuits and Systems for Video Technology. 26. 1-1. 10.1109/TCSVT.2015.2416555.
8. Aldabbagh, Ghassan. (2022). Comprehensive Review on Visual Tracking Systems. 2457-1016.
9. Li, Qihe & Luo, Yuping & Xiao, Deyun. (2006). Video Object Contour Tracking Using Improved Dual-Front Active Contour. 855-865. 10.1007/978-3-540-37275-2\_107.



10. Liu, Kangcheng & Chen, Xiao-Fan & Bo, Chun-Juan & Wang, Dong. (2022). Long-term Visual Tracking: Review and Experimental Comparison. *Machine Intelligence Research*. 19. 1-19. 10.1007/s11633-022-1344-1.
11. Zhao, Xin & Hu, Shiyu & Wang, Yipei & Zhang, Jing & Hu, Yimin & Liu, Rongshuai & Ling, Haibin & Li, Yin & Li, Renshu & Liu, Kun & Li, Jiadong. (2023). BioDrone: A Bionic Drone-Based Single Object Tracking Benchmark for Robust Vision. *International Journal of Computer Vision*. 10.1007/s11263-023-01937-0.
12. Kumar, Ashish & Jain, Rachna & Devi, Ajantha & Nayyar, Anand. (2023). Object Tracking Technology Trends, Challenges and Applications. 10.1007/978-981-99-3288-7.
13. Yu, Tao & Lin, Ne & Zhang, Xu & Pan, Yanqi & Hu, Huiyi & Zheng, Wenfang & Liu, Jiquan & hu, Weiling & Duan, Huilong & Si, Jianmin. (2022). An end-to-end tracking method for polyp detectors in colonoscopy videos. *Artificial Intelligence in Medicine*. 131. 102363. 10.1016/j.artmed.2022.102363.
14. Li, Xi & Hu, Weiming & Shen, Chunhua & Zhang, Zhongfei & Dick, Anthony & Hengel, Anton. (2013). A Survey of Appearance Models in Visual Object Tracking. *ACM Transactions on Intelligent Systems and Technology (TIST)*. 4. 10.1145/2508037.2508039.
15. Memarmoghadam, Alireza. (2021). The Eighth Visual Object Tracking VOT2020 Challenge Results.
16. Kang, Bin & Zhu, Wei-Ping & Liang, Dong. (2017). Robust multi-feature visual tracking via multi-task kernel-based sparse learning. *IET Image Processing*. 11. 10.1049/iet-ipr.2016.1062.
17. Dargazany, Aras & Soleimani, Ali & Ahmadyfard, Alireza. (2010). Multibandwidth Kernel-Based Object Tracking. *Adv. Artificial Intellegence*. 2010. 10.1155/2010/175603.
18. Baheti, Bhakti & Baid, Ujjwal & Talbar, Sanjay. (2016). An approach to automatic object tracking system by combination of SIFT and RANSAC with mean shift and KLT. 254-259. 10.1109/CASP.2016.7746175.

19. Ross, David & Lim, Jongwoo & Lin, Ruei-Sung & Yang, Ming-Hsuan. (2008). Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*. 77. 125-141. 10.1007/s11263-007-0075-7.
20. Shah, Syed Hammad Hussain & Karlsen, Anniken & Solberg, Mads & Hameed, Ibrahim. (2023). An efficient and lightweight multiperson activity recognition framework for robot-assisted healthcare applications. *Expert Systems with Applications*. 241. 122482. 10.1016/j.eswa.2023.122482.
21. Bevzyuk, Roman & Lyashenko, Oleksii. (2023). СИСТЕМА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДІВ З ВИКОРИСТАННЯМ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ. *Системи управління, навігації та зв'язку. Збірник наукових праць*. 2. 68-74. 10.26906/SUNZ.2023.2.068.
22. Ahamed, B. & V., Dharani & Poonkodi, M. & Sangeetha, G. & B., Sanaa. (2023). TRANSFER LEARNING AND DOMAIN ADAPTATION IN DEEP NETWORKS. 10.47715/JPC.V.978-93-91303-90-7.
23. Борківський, Б & Теслюк, В. (2023). Використання нейромережових засобів для розпізнавання об'єктів у мобільних системах з обходом перешкод. *Scientific Bulletin of UNFU*. 33. 84-89. 10.36930/40330412.
24. Ovalle-Magallanes, Emmanuel & Avina-Cervantes, Juan Gabriel & Cruz-Aceves, Ivan & Ruiz-Pinales, José. (2022). Improving Convolutional Neural Network Learning based on a Hierarchical Bezier Generative Model for Stenosis Detection in X-ray images. *Computer Methods and Programs in Biomedicine*. 219. 1-12. 10.1016/j.cmpb.2022.106767.
25. Кушнір, Надія & Локтікова, Тамара & Морозов, Андрій & Юрченко, Владислав. (2022). Використання згорткових нейронних мереж у задачах розпізнавання та класифікації об'єктів зображень. *Технічна інженерія*. 93-100. 10.26642/ten-2022-1(89)-93-100.
26. Dollar, Piotr & Wojek, Christian & Schiele, Bernt & Perona, Pietro. (2011). Pedestrian Detection: An Evaluation of the State of the Art. *IEEE transactions on pattern analysis and machine intelligence*. 34. 743-61. 10.1109/TPAMI.2011.155.

27. МАЛЬЦЕВ, АНТОН. (2022). МЕТОДИ МАШИННОГО НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПЕРЕДБАЧЕННЯ ВЕЛИКИХ ЗАШУМЛЕНИХ ДАНИХ ЗА ДОПОМОГОЮ СУЧАСНИХ МОВ ПРОГРАМУВАННЯ. Information Technology: Computer Science, Software Engineering and Cyber Security. 39-43. 10.32782/IT/2022-1-6.

28. Kristan, Matej & Leonardis, Ales & Matas, Jiri & Felsberg, Michael & Pflugfelder, Roman & Čehovin Zajc, Luka & Vojir, Tomas & Bhat, Goutam & Lukežič, Alan & Eldesokey, Abdelrahman & Fernandez Dominguez, Gustavo & Garcia-Martin, Alvaro & Iglesias-Arias, Álvaro & Alatan, A. & Gonzalez-Garcia, Abel & Petrosino, Alfredo & Memarmoghadam, Alireza & Vedaldi, Andrea & Muhič, Andrej & He, Zhiqun. (2019). The Sixth Visual Object Tracking VOT2018 Challenge Results. 10.1007/978-3-030-11009-3\_1.

29. Ісак, Людмила & Якуба, Василь & Бабак, Олександр. (2023). ЦИФРОВІ ІНСТРУМЕНТИ GOOGLE. Grail of Science. 219-224. 10.36074/grail-of-science.07.07.2023.034.

30. Athab, Suha & Karim, Abdulamir & Nermend, Kesra. (2023). A New Perspective for Mining COCO Dataset. 10.33103/uot.ijccce.23.3.7.

# ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

## (Презентація)



НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення



### МАГІСТЕРСЬКА РОБОТА

## «Розробка методу відстеження траєкторії людини в натовпі на основі технологій комп'ютерного зору»

Виконав: студент групи ПДМ-62, Усіченко Ангеліна Андріївна

Керівник: к.т.н., доцент Негоденко Олена Василівна

Київ - 2024

### МЕТА, ОБ'ЄКТ, ПРЕДМЕТ ДОСЛІДЖЕННЯ

**Мета роботи:** підвищення ефективності відстеження та формування відповідних траєкторій переміщення осіб за допомогою технологій комп'ютерного зору.

**Об'єкт дослідження:** відстеження траєкторії людей у натовпі на основі відеоматеріалів.

**Предмет дослідження:** застосування технологій комп'ютерного зору для фіксації траєкторії відстежуваних осіб.

## ВІЗУАЛЬНЕ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ

Проблеми візуального відстеження об'єктів

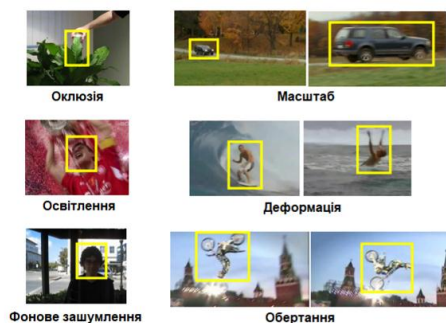
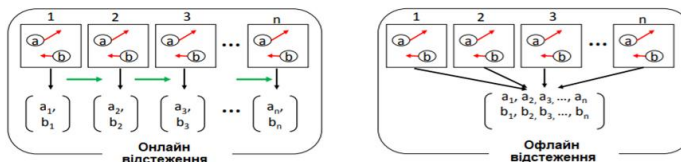


Схема відстеження онлайн та офлайн



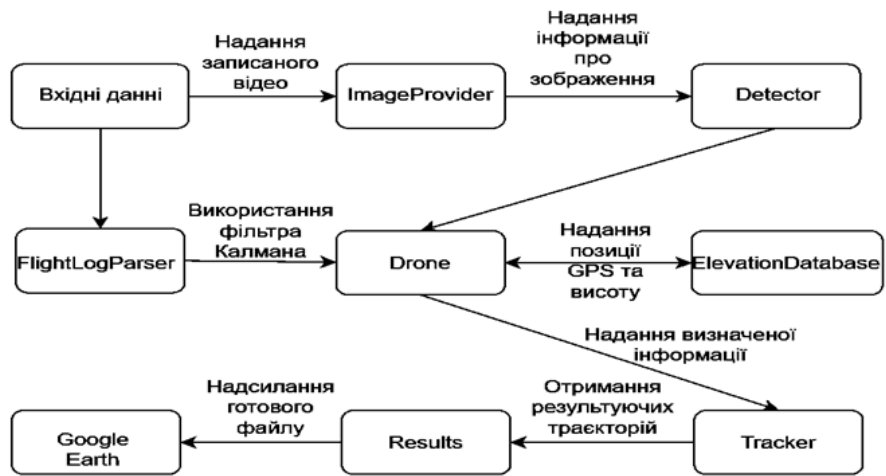
3

## ПОРІВНЯННЯ ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК ЗАСОБІВ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ

Метод відстеження	Кількість цілей		Тип інсталяції		Режим процесу		Тип моделі		Особливість	
	Одичні	Множинні	На основі моделі	Без моделі	Офлайн	Онлайн	Генеративні	Дискримінативні	Мануальні	Глибокі
<i>STRUCK</i>	+	-	-	+	-	+	-	+	+	-
<i>SCM</i>	+	-	-	+	-	+	-	+	+	-
<i>LSK</i>	+	-	-	+	-	+	+	-	+	-
<i>CF2</i>	+	-	-	+	-	+	-	+	-	+
<i>CIT</i>	-	+	+	-	-	+	-	+	+	-
<i>CSK</i>	-	+	-	+	-	+	-	+	+	-
<i>KMS</i>	-	+	-	+	-	+	+	-	+	-
<i>SBT</i>	-	+	-	+	-	+	-	+	+	-
<i>CT</i>	-	+	-	+	-	+	-	+	+	-
<i>IVT</i>	-	+	+	-	-	+	+	-	+	-
<i>MDNet</i>	+	-	-	+	-	+	-	+	-	+
<i>MTT</i>	+	-	-	+	-	+	+	-	+	-
<i>MTMVT</i>	+	-	-	+	-	+	+	-	+	-

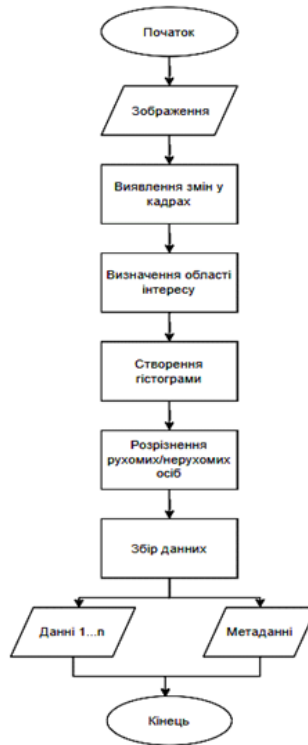
4

## СХЕМА ПРОЦЕСУ ВІДСТЕЖЕННЯ ТРЕКТОРІЇ ЛЮДИНИ В НАТОВПІ



5

## СХЕМА АЛГОРИТМУ ВИЯВЛЕННЯ ОСІБ НА ЗОБРАЖЕННІ



6

## РОЗРАХУНОК ВЕКТОРА НАПРЯМКУ ОСОБИ

Розрахунок висоти і ширини зображення:

$$\tan\left(\frac{fov_{horizontal}}{2}\right) = \frac{width}{distance};$$

$$\tan\left(\frac{fov_{vertical}}{2}\right) = \frac{height}{distance}.$$

Розрахунок ширини/висоти зрізу поля зору камери:

$$ratio = \frac{width}{height}.$$

Обчислення різниці між вектором напрямку огляду камери та вектором напрямку людини:

$$angle_{horizontal} = \frac{person_y - center_y}{height \cdot fov_{vertical}}$$

$$angle_{horizontal} = \frac{person_x - center_x}{width \cdot fov_{horizontal}}$$

Остаточний розрахунок тривимірного вектора напрямку для окремої людини в системі координат дрона:

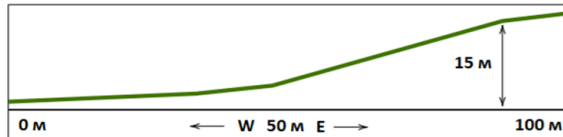
$$\vec{v} = distance \cdot [1, \tan(angle_{horizontal}), \tan(angle_{vertical})].$$



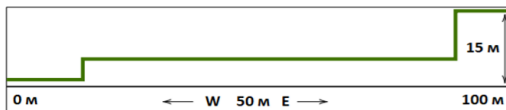
7

## ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ МЕТОДУ

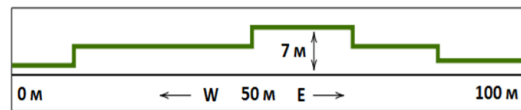
Фактичний профіль висоти (відповідно до Google Earth)



Профіль рельєфу за цифровою базою даних «The TanDEM-X 90m Digital Elevation Model»



Профіль рельєфу за цифровою базою даних «Copernicus Land Monitoring Service–EU-DEM»



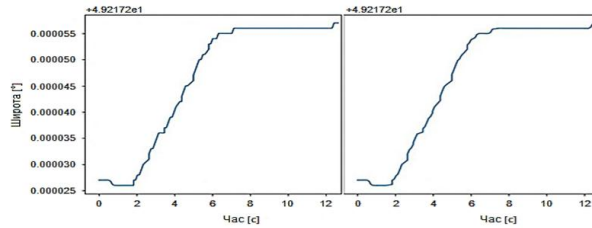
Приклад змін у виявленні людини після виконання коригування зображення

Метод	Параметри	Пропущено	Додано	Вище (сумарно)	Нижче (сумарно)
<i>Gauss Blur</i>	<i>ksize 5x5</i>	8	1	21 (1,60)	56 (-3,02)
<i>Bilateral Filter</i>	<i>sigmaColor 30, sigmaSpace 30</i>	6	1	27 (1,12)	49 (-1,68)
<i>CLAHE</i>	<i>clipLimit 1, tileGridSize 7x7</i>	1	1	33 (1,57)	40 (-0,99)

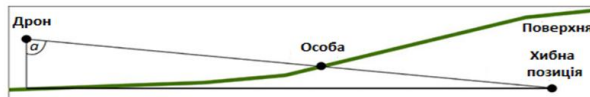
8

# ВИЗНАЧЕННЯ GPS-ПОЛОЖЕННЯ ЛЮДИНИ

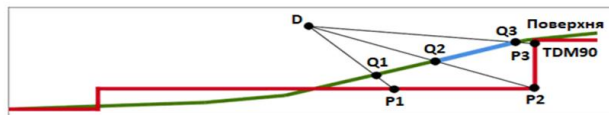
Застосування фільтра Калмана до широти GPS-положення дрона з тестового набору



Приклад неправильного визначення GPS-положення людини, коли актуальне значення висоти є невідомим



Варіативність виявлення осіб, залежно від перепаду висот поверхні



# АНАЛІЗ РЕЗУЛЬТАТІВ ОБРОБКИ ДАНИХ

Візуалізація траєкторій на Google Earth у форматі GPX



Візуалізація траєкторій на Google Earth у форматі KML



Необроблена траєкторія особи (56 метрів за 10 секунд)



Результуюча траєкторія особи (агрегована за інтервали 0,5 с і згладжена за допомогою фільтра Калмана)





## **ВИСНОВКИ**

1. Проведено аналіз функціональних характеристик існуючих методів та інструментів виявлення людей засобами комп'ютерного зору, з метою подальшого формування відповідних траєкторій руху.
2. Розглянута побудова та функціонування згорткових нейронних мереж, їх використання для виявлення людей на зображенні та можливі варіації, придатність яких порівнюється в контексті обробки відеозапису, знятого з дрона, тобто при перегляді з висоти і з різних ракурсів.
3. Запропоновано методи визначення GPS-координат відстежуваних осіб без використання спеціального обладнання для дронів. Також описано загальні принципи методів повторної ідентифікації осіб та конкретні шляхи реалізації відповідних задач.
4. Випробувано 4 різних варіанти детекторів, та проведено аналіз ефективності їх застосування для різних ситуацій. Для отриманих експериментів було обрано раціональний варіант, який на тестовому наборі даних досягає високої точності.
5. Сформовано набір ідентифікаторів, що містить 384 зображення та 5 відповідних осіб. Де обробка даних виконувалась не лише базовими методами повторної ідентифікації за допомогою гістограм, а й за допомогою машинного навчання, а саме нейронної мережі AlignedReID

**ДЯКУЮ ЗА УВАГУ!**