

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РОЗРОБЛЕННЯ МЕТОДУ ДЕТЕКЦІЇ ТА
ТРЕКІНГУ ЛЮДЕЙ В ЗАДАНІЙ ОБЛАСТІ ДЛЯ СИСТЕМ КОНТРОЛЮ

НАВАНТАЖЕННЯ

(тема)

Виконав:

студент 2 курсу, групи ІНФМ-20-1

Подвальний Є.С.

(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика

(повна назва освітньої програми)

Керівник Любченко Валентин Анатолійович

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Кобилін О.А.

(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Подвальному Євгенію Сергійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження та розроблення методу детекції та трекінгу людей в заданій області для систем контролю навантаження

затверджена наказом по університету від « 22 » жовтня _____ 2021 року № 1574 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 2021 р.

3. Вихідні дані до роботи Теоретичні відомості про розпізнавання об'єктів; Методи детектування осіб в відеопотоці; Алгоритми трекінгу; Згорткові нейронні мережі; Перелік використовуваних програмних засобів реалізації алгоритмів трекінгу об'єктів.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд алгоритмів розпізнавання об'єктів.

2. Огляд трекінгу об'єктів .

3. Аналіз існуючих методів трекінгу.

4. Аналіз сучасних методів детектування.

5. Вибір оптимальних методів трекінгу для предметної області

6. Розробити власний алгоритм онлайн трекінгу множини об'єктів у режимі реального часу на основі вибраних методів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність задачі, методи детектування, алгоритми трекінгу, приклади роботи системи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Белова Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	22.10.2021	виконано
2	Аналіз завдання, підбір літератури	23.10.21-07.11.21	виконано
3	Аналіз літератури з досліджуваної проблеми	26.10.21-10.11.21	виконано
4	Аналіз технічних засобів	01.11.21-07.11.21	виконано
5	Розробка методу	07.11.21-14.11.21	виконано
6	Програмна реалізація	14.11.21-17.11.21	виконано
7	Оформлення пояснювальної записки	17.11.21-22.11.21	виконано
8	Перевірка на плагіат	03.12.2021	
9	Рецензування	04.12.2021	
10	Підготовка презентації та доповіді	06.12.2021	
11	Занесення роботи в електронний архів	08.12.2021	
12	Попередній захист кваліфікаційної роботи	08.12.2021	

Дата видачі завдання 22 жовтня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Любченко В.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 70 с., 3 табл., 28 рис., 1 дод., 42 джерело.

ДЕТЕКТУВАННЯ ОБ'ЄКТІВ, ТРЕКІНГ ОБ'ЄКТІВ, КОМП'ЮТЕРНИЙ ЗІР.

Об'єктом дослідження є послідовність різноракурсних кадрів відеопотоку.

Предметом дослідження є детекція та трекінг множини об'єктів на кадрах відеопотоку.

Метою дослідження є аналіз та розробка алгоритмів онлайн детекції та трекінгу множини об'єктів у режимі реального часу на кадрах відеопослідовності.

Проведено дослідження алгоритмів розпізнавання об'єктів та методів трекінгу. Досліджено сучасні алгоритми трекінгу та детекції.

У результаті роботи здійснена програмна реалізація онлайн детекції та трекінгу множини об'єктів на основі обраних методів, що дозволяє працювати у режимі реального часу, а також на записаних відеофайлах.

OBJECT DETECTION, OBJECT TRACKING, COMPUTER VISION.

The object of the study is a sequence of different frames of the video stream.

The subject of the study is the detection and tracking of many objects in the frames of the video stream.

The aim of the study is analysis and development algorithms for online detection and tracking of many objects in real time on video frames.

A study of object recognition algorithms and tracking methods. Modern tracking and detection algorithms are studied.

As a result of implemented software implementation of online detection and tracking of many objects based on selected methods, which allows you to work in real time, as well as on recorded video files.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Сучасний стан питання детекції об'єктів	9
1.1 Розпізнавання об'єктів	9
1.2 Огляд алгоритмів розпізнавання об'єктів	10
1.2.1 Метод гнучкого порівняння на графах	11
1.2.2 Нейронні мережі.....	13
1.2.3 Приховані Марксівські моделі.....	15
1.2.4 Метод головних компонент	15
1.3 Трекінг об'єктів.....	16
1.3.1 Поняття та задача трекінгу.....	17
1.3.2 Трекери на основі кореляційних фільтрів	18
1.3.3 Трекери на основі детектування.....	19
1.3.4 Трекери на основі графів.....	19
1.3.5 Трекери на основі нейронних мереж	20
1.3.6 Існуючі методи трекінгу.....	21
1.3.6.1 Методи передбачення.....	23
1.3.6.2 Методи вилучення ознак.....	23
1.3.6.3 Методи розрахунку метрик схожості	24
1.3.6.4 Методи асоціації	24
1.3.7 Сучасні методи детектування	25
1.4 Постановка задачі дослідження.....	26
2 Математичні моделі детекції й трекінгу об'єктів.....	27
2.1 Огляд моделі детекції та трекінгу	27
2.1.1 Задача детекції	27
2.1.2 Математична модель детекції.....	28
2.1.3 Проблематика детекції	30
2.1.6 Математична модель трекінгу	31

2.2	Методи оцінки якості роботи алгоритму трекінгу	
	множини об'єктів	35
2.2.1	Методи оцінки якості роботи методу детекції.....	35
2.2.2	Методи оцінки якості роботи алгоритму трекінгу	36
2.3	Розробка архітектури алгоритму трекінга.....	37
2.4	Розробка методів детектування	40
2.5	Розробка методів передбачення руху для алгоритму трекінга	43
2.5.1	Використання оптичного потоку	43
2.5.2	Використання вектора швидкості	44
2.5.3	Використання фільтра Калмана	44
2.6	Розробка методів вилучення графічних ознак для алгоритму трекінга.....	47
2.7	Розробка методів обчислення метрик схожості для алгоритму трекінга.....	47
2.7.1	Використання Intersection Over Union	48
2.7.2	Використання косинусної схожості.....	48
2.8	Розробка методів асоціації.....	49
2.9	Розробка методів оцінки на основі результатів алгоритма трекінгу	50
3	Комп'ютерна модель.....	53
3.1	Обґрунтування вибору програмної реалізації	53
3.2	Обґрунтування вибору метода детектування для алгоритму.....	59
3.3	Тестування розробленої системи	57
3.4	Ілюстрація роботи системи.....	59
	Висновки	65
	Перелік джерел посилання	65
	Додаток А Блок-схеми роботи алгоритму	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Детекція – процес локалізації об'єктів на зображенні

Трек – набір положень об'єкта на кожному кадрі відеопослідовності

Трекінг – процес відстеження об'єкта на відеопослідрвності, визначення траєкторії об'єкта

Обмежуюче вікно (bounding box) – проста фігура (прямокутник), що обмежує форму більш складнішої геометричної фігури

VOT – Visual Object Tracking

MOT – Multiple Object Tracking

ПММ – приховані Марковські моделі

HMM – Hidden Markov Model

PCA – Principal Component Analysis

CNN – Convolutional Neural Network

ЕОМ – електронна обчислювальна машина

IOU – Intersection Over Unoin

CV – Computer Vision

ВСТУП

Розпізнавання осіб, як одне з найбільш успішних програм аналізу зображень, останнім часом привернула до себе велику увагу. Це пов'язано з наявністю реальних технологій, в тому числі мобільних рішень. Аналіз в області автоматичного розпізнавання осіб ведуться з 1960-х років, але проблема все ще залишається невирішеною. Здовж останнього десятиріччя забезпечило важливий прогрес в цій області за допомогою досягнень в області моделювання і аналізу осіб. Не зважаючи на те, що системи були розроблені для виявлення та відстеження осіб, надійне розпізнавання осіб і у минулому вело за собою серйозну проблему для дослідників комп'ютерного зору і розпізнавання образів. Існує декілька причин зросту останнім часом інтересу до розпізнавання осіб, в тому числі зростаюча збентеженість суспільства безпекою, необхідна перевірка особи в цифровому світі, методи аналізу і моделювання осіб в управлінні мультимедійними даними.

Успіхи в області автоматизованого аналізу осіб, розпізнавання образів і машинного навчання дозволили розробити системи автоматичного розпізнавання осіб для вирішення цих завдань. З одного боку, розпізнавання особи – природний процес, через те що люди зазвичай роблять це без особливих зусиль і без особливого усвідомлення. З іншого боку, вживання цього процесу в області комп'ютерного зору залишається складною проблемою. Будучи частиною біометричної технології, автоматичне розпізнавання осіб має безліч бажаних властивостей. В їх основі лежить важлива перевага – неінвазійність.

1 СУЧАСНИЙ СТАН ПИТАННЯ ДЕТЕКЦІЇ ОБ'ЄКТІВ

Завдання детекції об'єктів має безліч застосувань у таких галузях, як організація відеоконференцій, системи машинного зору, системи безпеки та контролю доступу. Основною складністю даної задачі є залежність якості результату розпізнавання людини за зображенням особи від ракурсу, положення, умов освітлення, наявності деформації, прихованих частин, поворотів щодо осей та інших факторів, що впливають на якість розпізнавання образів. Далі будуть розглянуті та проаналізовані сучасні методи розпізнавання осіб.

1.1 Розпізнавання об'єктів

Продуктивність розпізнавання об'єктів є ключовою проблемою, тому обговорюються методи роботи з нефронтальним розпізнаванням осіб. Моделювання підпростору і методи зменшення розмірів на основі навчання є фундаментальними за для багатьох сучасних методів розпізнавання осіб. Виявлення таких підпросторів для вилучення ефективних функцій і побудови надійних класифікаторів є ще одною проблемою в цій галузі. Розпізнавання осіб має гідності як високої точності, так і низькою нав'язливості.

Початковий етап – це виявлення осіб на отриманому зображенні незалежно від масштабу та місця розташування. Він часто використовує розширену процедуру фільтрації, щоб розрізнити розташування, що представляють особи, і фільтрувати їх за допомогою точних класифікаторів. Примітно, що всі зміни зміщення, масштабування і повороту повинні оброблятися на етапі виявлення особи.

На наступному етапі відбувається аналіз обличчя виявленої особи, яка заснована на наборі антропометричних даних та прогнозує приблизне розташування основних функцій, таких як очі, ніс і рот. Звичайно, вся

процедура повторюється для прогнозування підфункцій щодо основних функцій і перевіряється статистикою колокацій, щоб відхилити будь-які неправильно розташовані функції.

Виділені опорні точки генеруються в результаті геометричних комбінацій на зображенні особи, а потім починається фактичний процес розпізнавання. Це виконується шляхом знаходження локального подання зовнішнього вигляду обличчя в кожній з опорних точок. Схема уявлення залежить від підходу.

1.2 Огляд алгоритмів розпізнавання об'єктів

Існуючі основні методи розпізнавання (рис. 1.1)[1]:

- метод гнучкого порівняння на графах;
- нейронні мережі;
- скриті Марківські моделі;
- метод головних компонент.

Основною відмінністю алгоритмів буде обчислення ознак та порівняння їхніх сукупностей між собою [2].

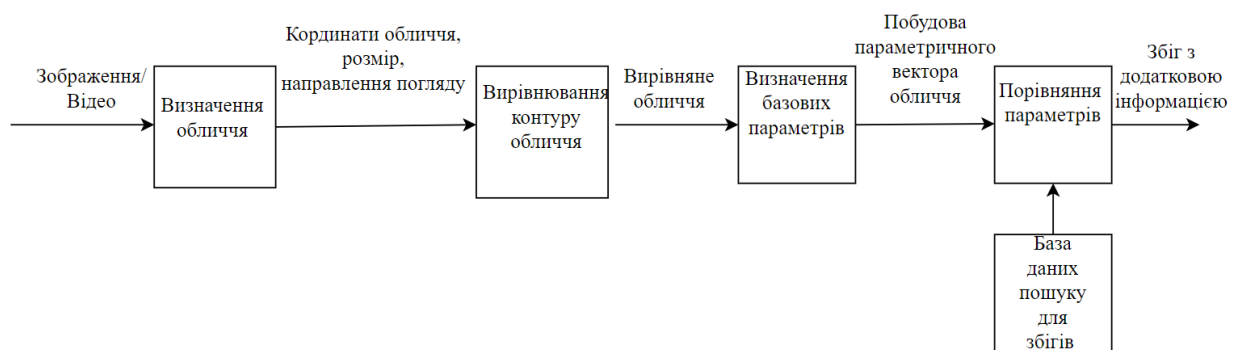


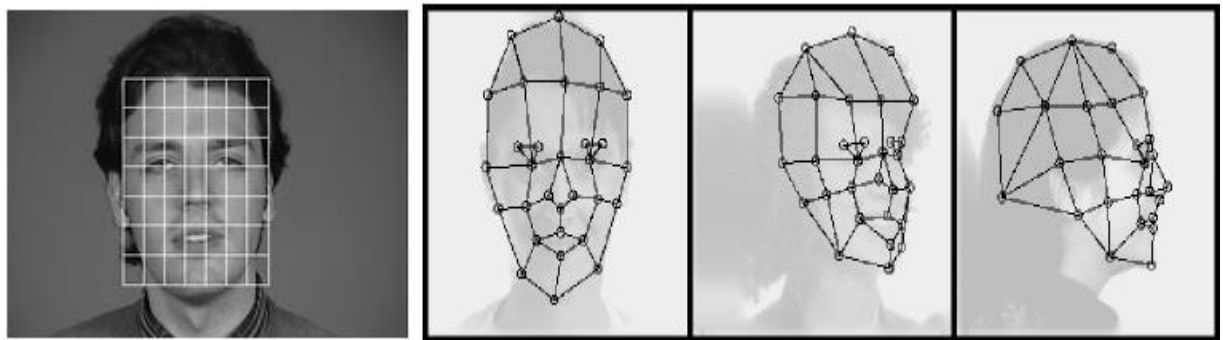
Рисунок 1.1 – Загальний процес розпізнавання

Іноді системи розпізнавання неспроможні точно визначити об'єкти. Зазвичай помилка виникає через погану якість зображення або недостатню

інформацію. Негативне освітлення або низька якість зображення можуть ускладнити точність аналізу вузлових точок. Наприклад, на дані може вплинути затемнення ряду рис обличчя. Це створює помилку, внаслідок чого неможливо буде порівняти його з правильними даними.

1.2.1 Метод гнучкого порівняння на графах

Суть методу зводиться до еластичного порівняння графів, які описують зображення осіб. Особи представлені у вигляді графів зі зваженими вершинами та ребрами (рис. 1.2) [2]. На етапі розпізнавання один із графів: еталонний – залишається незмінним, тоді як інший деформується з метою найкращого припасування до першого. У подібних системах розпізнавання графи можуть являти собою як прямокутні ґрати, так і утворену характерними (антропометричними) точками структуру.



а)

б)

Рисунок 1.2 – Структура графа для розпізнавання осіб:

а) регулярні ґрати;

б) графік на основі антропометричних точок обличчя

У вершинах графа обчислюються значення ознак, найчастіше використовують комплексні значення фільтрів Габора чи його впорядкованих наборів Габора показано на рисунку 1.3, які обчислюються у деякій локальній

області вершини графа локально шляхом згортки значень яскравості пікселів з фільтрами Габора [2].

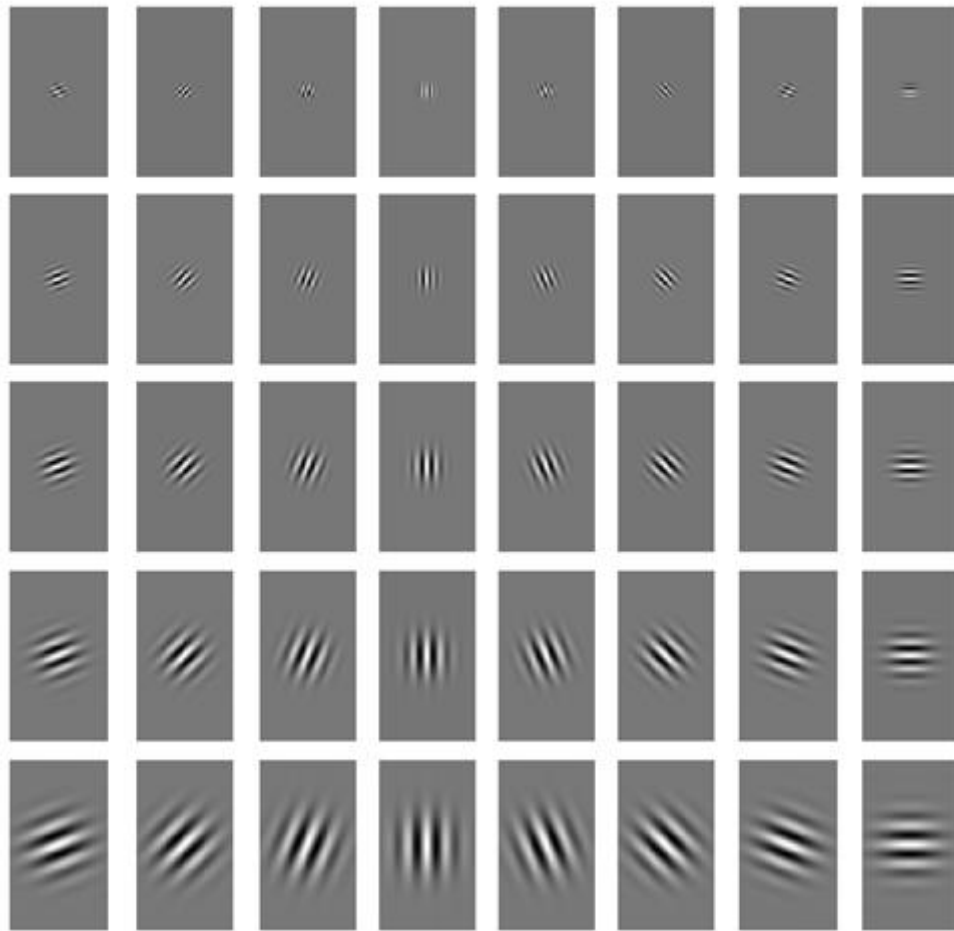


Рисунок 1.3 – Набор фільтрів Габора

Ребра графа зважуються відстанями між суміжними вершинами. Відмінність між двома графами обчислюється за допомогою деякої цінової функції деформації, яка враховує як різницю між значеннями ознак, обчисленими у вершинах, і ступінь деформації ребер графа.

Деформація графа відбувається шляхом зміщення кожної з його вершин на деяку відстань у певних напрямках щодо її вихідної розташування та вибору такої її позиції, при якій різниця між значеннями ознак у вершині графа, що деформується, і відповідної їй вершині еталонного графа буде мінімальною. Ця операція виконується по черзі всім вершин графа до того часу, доки досягнуто найменше сумарне різницю між ознаками деформуємого і

еталонного графів. Значення цінової функції деформації при такому положенні графа, що деформується, і буде мірою відмінності між вхідним зображенням особи і еталонним графом. Ця процедура деформації (рис. 1.4), має виконуватися для усіх еталонних осіб, закладених у базу даних системи. Результат розпізнавання системи – стандарт з найкращим значенням цінової функції деформації [2].

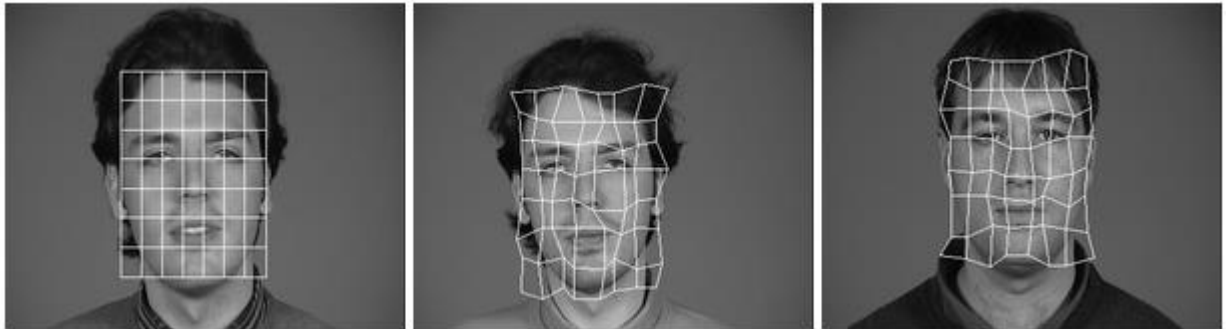


Рисунок 1.4 – Приклад деформації графа у вигляді регулярних ґрат

Недоліком даного методу є висока обчислювальна складність процедури розпізнавання, що викликає лінійну залежність часу роботи від обсягу бази даних осіб, а також низька технологічність при запам'ятовуванні нових стандартів.

1.2.2 Нейронні мережі

Найбільш поширеним варіантом є використання мережа, побудованої на багат шаровому перцептроні (рис. 1.5), що дозволяє класифікувати подане на вхід зображення відповідно до попереднього налаштування мережі [3].

Навчаються нейронні мережі з набору навчальних прикладів. Мета навчання зводиться до налаштування міжнейронних зв'язків у процесі розв'язання оптимізаційної задачі методом градієнтного спуску. У процесі навчання нейронної мережі відбувається автоматичне вилучення ключових

ознак, визначення їх важливості та побудова взаємозв'язків між ними. Передбачається, що навчена нейронна мережа зможе застосувати досвід, отриманий у процесі навчання, на невідомі образи за рахунок узагальнюючих здібностей.

Найкращі результати у сфері розпізнавання осіб показала згорткова нейронна мережа яка є логічним розвитком таких ідей архітектури як когнітрон та неокогнітрон. Успіх обумовлений можливістю обліку двовимірної топології зображення, на відміну багат шарового перцептрона.

Відмінними рисами згорткової нейронної мережі є локальні рецепторні поля, які забезпечують локальну двовимірну зв'язність нейронів, загальні ваги забезпечують детектування деяких рис у будь-якому місці зображення та ієрархічна організація з просторовими вибірками.

Завдяки цим нововведенням згорткової нейронної мережі забезпечує часткову стійкість до змін масштабу, зсувів, поворотів, зміни ракурсу та інших спотворень.

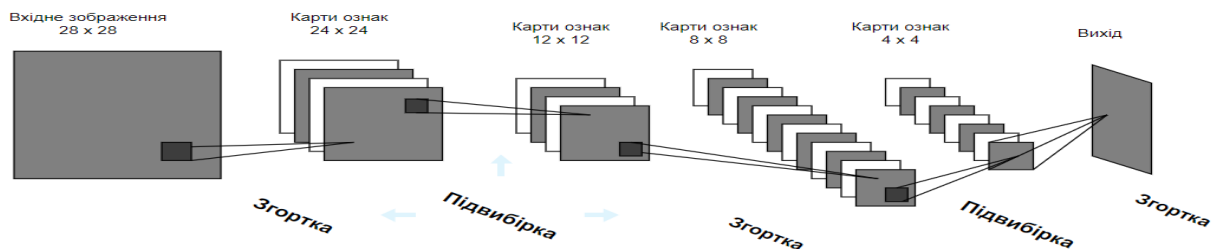


Рисунок 1.5 – Принцип роботи згорткових нейронних мереж

До недоліків нейронних мереж можна віднести додавання нової еталонної особи до бази даних, яке вимагає повного перенавчання мережі по всьому наявному наборі, тобто є досить тривалою процедурою, залежно від розміру вибірки. Проблеми математичного характеру, пов'язані з навчанням: потрапляння в локальний оптимум, вибір оптимального кроку оптимізації, перенавчання. Можна зробити висновок, що нейронна мережа – «чорна скринька» з результатами роботи, що важко інтерпретуються.

1.2.3 Приховані Марківські моделі

Одним із статистичних методів розпізнавання образів є приховані Марківські моделі (ПММ) з дискретним часом. ПММ використовують статистичні властивості сигналів та враховують безпосередньо їх просторові характеристики. Елементами моделі є: безліч прихованих станів, безліч станів, що спостерігаються, а саме матриця перехідних ймовірностей, початкова ймовірність станів. Кожному відповідає своя Марківська модель. При розпізнаванні об'єкта перевіряються згенеровані для заданої бази об'єктів Марківські моделі і шукається максимальна ймовірність того, що послідовність спостережень для даного об'єкта згенерована відповідною моделлю.

До недоліків віднесемо необхідність підбирати параметри моделі для кожної бази даних. А також те, що ПММ не володіє здатністю розрізняти, тобто алгоритм навчання тільки максимізує відгук кожного зображення на свою модель, але не мінімізує відгук на інші моделі.

1.2.4 Метод головних компонент

Метод головних компонент (РСА), заснований на перетворенні Карунена-Лоева.

Головною метою методу головних компонентів є значне зменшення розмірності простору ознак таким чином, щоб воно якнайкраще описувало «типові» образи, що належать множині осіб [2]. Використовуючи цей метод, можна виявити різні мінливості в навчальній вибірці зображень осіб і описати цю мінливість у базисі кількох ортогональних векторів, які називаються власними.

Отриманий один раз на навчальній вибірці зображень осіб набір власних векторів використовується для кодування решти зображень осіб, які

видаються зваженою комбінацією цих власних векторів. Використовуючи обмежену кількість власних векторів можна отримати стислу апроксимацію вхідного зображення особи, яку потім можна зберігати в базі даних у вигляді вектора коефіцієнтів, що є одночасно ключем пошуку в базі даних осіб.

Суть методу основних компонентів зводиться до того, що спочатку весь навчальний набір осіб перетворюється на одну загальну матрицю даних, де кожен рядок є один екземпляр зображення особи, розкладеного в рядок. Усі особи навчального набору повинні бути приведені до одного розміру та з нормованими гістограмами.

1.3 Трекінг об'єктів

Трекінг – одна з найпоширеніших та найважливіших проблем у сфері громадського спостереження. Це визначається необхідністю знати, де знаходяться об'єкти у кожен момент аналізованого відео. Класичні алгоритми виявлення не вирішують цієї проблеми.

Коли ми застосовуємо алгоритм виявлення об'єкта до першого кадру аналізованого відеофрагменту, визначається присутність об'єкта та його положення на кадрі, або просто виділяється область, що цікавить. Завдання трекінгу, своєю чергою, – знайти становище обраного об'єкта у наступних кадрах.

Основні труднощі, з якими можуть зіткнутися алгоритми відстеження – це те, що трекер може легко переключитися на інший об'єкт під час відстеження або втратити об'єкт, який необхідно відстежувати, масштабування, поворот об'єкта, зміна освітлення, розмиття руху. Класичні алгоритми націлені на вирішення цієї проблеми, але часто вони цього не роблять. Це відбувається через труднощі адаптації та налаштування цих алгоритмів. Алгоритми на основі нейронних мереж, у свою чергу, краще

справляються із завданням відстеження, але їх справді складно інтерпретувати та контролювати.

1.3.1 Поняття та задача трекінгу

У більшості випадків одна й та сама людина знаходиться в полі зору камери відеоспостереження протягом певного часового інтервалу, тому його зображення можна знайти в послідовності кадрів. Завдяки цьому можна здійснювати трекінг шуканої особи. У даному випадку, трекінг (tracking) – це визначення місцезнаходження об'єкта, що рухається, у відеопотоці. В алгоритмі використовується алгоритм, який аналізує кадри відеопотоку, і видає положення цільових предметів, що рухаються щодо кадру, що дозволяє скласти послідовність зображень конкретного об'єкта.

Об'єкт, що рухається, на відео класифікується на регулярний (рух об'єкта по безперервній лінії), нерегулярний (розриви траєкторії внаслідок перекриття об'єкта стеження іншими об'єктами). У разі регулярної їзди об'єкта, трекер може розглядатися як динамічна система.

Отже, завдання трекінгу об'єкта визначається як завдання автоматичного розпізнавання та подальшого відстеження об'єкта на послідовності кадрів відеопотоку [4]. За допомогою алгоритму трекінга визначається траєкторія об'єкта в часі, шляхом визначення положення об'єкта на кожному кадрі відеопослідовності. Положення об'єкта на кадрі може бути представлене у вигляді точки або набору точок, простої геометричної фігури; силует або контур та іншої форми [5]. Подання положення об'єкта у вигляді обмежуючого вікна підходить для об'єктів будь-яких розмірів і форм і є однією з найчастіше використовуваних форм подання об'єкта на кадрі в існуючих рішеннях і наборах даних для навчання та валідації алгоритмів трекінгу.

Розрізняють трекінг одного об'єкта (Visual Object Tracking) та множини об'єктів (Multiple Object Tracking). Завдання трекінгу множини об'єктів полягає у одночасному детектуванні та трекінгу множини об'єктів на послідовності кадрів відеопотоку[4].

1.3.2 Трекери на основі кореляційних фільтрів

Основна ідея кореляційного трекінгу полягає у навчанні кореляційного фільта H за допомогою якого можна відокремити ціль від заднього фону на кадрі [6]. Параметри кореляційного фільтра підбирається таким чином, що його згортка із зображенням являла собою розподіл Гауса з піком у центрі об'єкта, що відстежується. Для навчання такого фільтра використовується навчальна вибірка, на якій мінімізується помилка між реальним результатом згортки та очікуваним, за допомогою методу найменших квадратів за формулою:

$$H = \arg \min \sum_{i=0}^n (F_i \odot H - G_i)^2, \quad (1.1)$$

де H – кореляційний фільтр;

F_i – вектори ознак;

G_i – функція розподілу Гауса.

У загальному вигляді для навчання фільтра (1.1) доводиться вирішувати систему рівнянь, що ускладнює обробку кадрів у часі.

Алгоритм трекінгу об'єкта за допомогою кореляційного трекера. Першим кроком з зображення витягують візуальні ознаки, які використовуються для навчання кореляційного фільтра. Через час після початку зйомки на перших кадрах по черзі відбувається процес локалізації цілі оновлення візуальної моделі за допомогою зображення, отриманого в новій позиції.

1.3.3 Трекери на основі детектування

Трекінг на основі детектування полягає у тому, що у кожному кадрі об'єкти виявляються детектором, та зв'язуються у траєкторії [6]. Щоб зв'язати об'єкти на двох сусідніх кадрах, спочатку для об'єктів з попереднього кадру передбачаються на поточному кадрі. Потім передбачені об'єкти зіставляються з об'єктами, які виявив детектор за допомогою ознак схожості (просторових чи візуальних). Траєкторії оновлюються з координатами відповідних зіставлених об'єктів, виявлених, детектором. Для непорівнянних об'єктів ініціалізуються нові траєкторії.

Використання даного методу дозволяє автоматично виявляти появу нових цілей в кадрі й зникнення старих, а також дозволяє коригувати параметри моделі за рахунок виправлення помилки між станом, отриманим з детектора та передбаченим станом.

1.3.4 Трекери на основі графів

Вхідними даними для трекера є k -дольний повний граф (рис. 1.6), де є ребро між кожною пари вузлів, що не входять до одного кластера. Для першого шару ділиться відео на сегмент з 10 кадрів і кожен сегмент визначає один кластер. Вузли графів у першому шарі є треклети низького рівня (середнього рівня треклети використовуються для створення входу до другого шару), які знайдено в кожному сегменті з максимальною довжиною 10 кадрів і мінімальна довжина 5 кадрів. Ці низькорівневі доріжки знаходяться за допомогою простого критерію перекриття, де обмежують коробки, які перекриваються у послідовних кадрах. У методі треклети не тільки допомагають зменшити складність обчислень, але також дозволяють включення подібності руху до вартості краю нашого графу. Тому що ребра нашого графа з'єднують більше двох гіпотез виявлення, які необхідні для

кодування вартості руху. Таким чином, кожному ребру в нашому графі, призначається вага, що включає як зовнішній вигляд, так і рух.

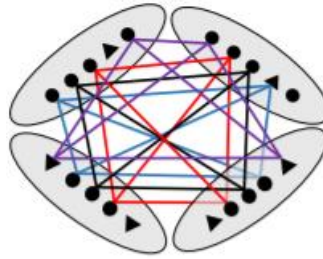


Рисунок 1.6 – Вхідний k -дольний граф

Після виявлення об'єктів другий крок спрямований на зв'язування різних екземплярів одного і того ж об'єкта з часом (тобто, асоціація даних). Типова проблема для асоціативних даних полягає в тому, щоб розрізнити предмети з подібним виглядом і рухом. З цієї причини асоціацію даних для відстеження об'єктів можна уподібнити проблемі відповідності руху. Основними недоліками методів є велика кількість параметрів, які необхідно налаштувати, що необхідні для моделювання простору станів [6]. Змінна кількість об'єктів дозволяється, коли фіктивні вузли вводяться в графік, щоб отримати постійне число вузлів на кадр.

1.3.5 Трекери на основі нейронних мереж

Останнім часом згорткові нейронні мережі дуже ефективні у сфері трекінгу, але, на жаль, високоточні алгоритми відстеження на основі CNN часто є занадто повільними для практичних систем [6]. Є лише кілька методів, які досягають двох потенційно суперечливих цілей – точності і швидкості – водночас. Алгоритми візуального відстеження на основі CNN зазвичай формулюють відстеження об'єктів як дискримінаційні проблеми виявлення об'єктів. Деякі методи малюють набір вибірок, що відповідають регіонам-кандидатам, і обчислюють їх ймовірність незалежно за допомогою CNN.

CNN попередньо підготовлені для виконання завдань класифікації зображень, мережі є конкурентоспроможними за для передбачення семантичної мітки зображень, але не чутливі до визначення відмінностей між екземплярами об'єктів у представленнях низького або середнього рівня. Пряме застосування такого CNN для візуального відстеження часто призводить до погіршення точності, оскільки вбудовування, створене попередньо підготовленими CNN для завдання класифікації зображень, не є ефективним для того, щоб розрізнити два об'єкти в одній категорії.

Трекер ефективно оцінює цільові стани на основі спостережень за допомогою ручної роботи, тоді як верифікатор двічі перевіряє оцінку за допомогою функції глибоких нейронних мереж.

1.3.6 Існуючі методи трекінгу об'єктів

На сьогоднішній день запропоновано багато алгоритмів і методів, що застосовуються у вирішенні задачі трекінгу множини об'єктів, не всі з яких підходять для трекінгу в режимі реального часу. Існують алгоритми, що реалізують мультиагентний підхід [5], який застосовує методи трекінгу одного об'єкта для кожного, що реалізують моделі для виконання всіх етапів обробки кадру та отримання результатів трекінгу множини об'єктів цілком (end-to-end) [7]. Застосування даних підходів для вирішення задачі онлайн трекінгу множини об'єктів у режимі реального часу можливе, але важко [8].

Стандартним підходом для вирішення задачі онлайн трекінгу в режимі реального часу є трекінг за допомогою детекцій [4]. При використанні даного підходу для кожного кадру відеопослідовності за допомогою алгоритму детектування визначається набір обмежувальних вікон об'єктів, тобто набір детекцій, і завданням алгоритму трекінгу є асоціація детекцій траєкторій об'єктів, що відстежуються [9].

Незважаючи на велику відмінність, для більшого числа алгоритмів можна виділити основні етапи вирішення задачі трекінгу иножини об'єктів за допомогою детекцій [10]:

- етап детектування об'єктів : на даному етапі відбувається аналіз поточного вхідного кадру для локалізації об'єктів певних класів; отриманий набір обмежувальних вікон, для кожного з яких зазначені ступеня належності класам;

- етап передбачення руху: на даному етапі для кожного відстежуваного об'єкта по зібраній на попередніх кадрах інформації (траєкторія, зображення тощо) передбачається рух, тобто знаходиться передбачуване обмежуюче вікно відстежуваного об'єкта на поточному вхідному кадрі;

- етап вилучення ознак: даний етап не завжди є частиною алгоритму трекінгу; на даному етапі відбувається витяг ознак з детекцій і передбачених обмежувальних об'єктів, що обстежуються; ці ознаки далі застосовуються для розрахунку метрик схожості;

- етап розрахунку метрик схожості: на даному етапі розраховуються метрики схожості для кожної пари передбаченого обмежувального вікна відстежуваного об'єкта і детекції;

- етап асоціації: для цього використовуються метрики схожості між передбаченими обмежуючими вікнами відстежуваних об'єктів і детекціями; траєкторією об'єктів, для яких детекція не була знайдена, можуть бути продовжені за допомогою передбачених вікон, що раніше обмежують; для детекцій, які були призначені об'єктам, створюються нові об'єкти.

1.3.6.1 Методи передбачення руху

Методи передбачення руху використовуються для передбачення обмежуючого вікна об'єкта, що відстежується на поточному кадрі за

інформацією, зібраною на попередньому кадрі. Одні з методів, що застосовуються на даному етапі роботи алгоритму трекінгу множини об'єктів:

- Оптичний потік [11, 12];
- Нейромережні моделі довгої короткострокової пам'яті (Long Short Term Memory, LSTM) [11, 13];
- фільтр Калмана [13 – 15].

1.3.6.2 Методи вилучення ознак

Методи вилучення ознак застосовуються для отримання ознак з обмежувальних вікон детекцій і передбачених об'єктів, що обмежують вікон. В алгоритмах трекінгу застосовуються два основні підходи для вилучення векторів графічних ознак з вікон, що обмежують:

- вилучення векторів ознак із зображення, що відповідає кожному обмежувальному вікну, за допомогою моделі згорткової нейронної мережі (Convolutional Neural Network, CNN) [13, 16];
- вилучення карти ознак із зображення з наступним зіставленням піднаборів цих ознак кожному обмежує вікну [17, 18].
- вилучення карти графічних ознак зображення чи обмежувальних вікон детекцій можливе на етапі роботи алгоритму детектування [15, 19].

У разі етап детектування об'єктів служить як локалізації обмежувальних вікон, але й отримання ознак. Інший часто використовуваний підхід до вилучення графічних ознак – використання для цього окремої моделі згорткової нейронної мережі [19].

1.3.6.3 Методи розрахунку метрик схожості

Метрики схожості, що обчислюються для кожної пари передбаченого обмежувального вікна об'єкта, що відстежується, і обмежує вікна детекції,

характеризують ймовірність того, що обидва обмежуючі вікна належать одному об'єкту. В алгоритмах трекінгу множини об'єктів часто застосовуються такі метрики схожості як IOU (Intersection Over Union) [14], косинусна подібність між векторами графічних ознак [7], ознаки руху (motion features) [14, 20] або форми (shape features) [20]. Також можливе застосування моделі сіамських нейронних мереж для обчислення метрики схожості між двома зображеннями, що обмежують вікна [21].

1.3.6.4 Методи асоціації

Методи асоціації застосовуються зіставлення відстежуваних об'єктів і детекцій з допомогою розрахованих раніше метрик схожості. Детекції, які не були призначені існуючим об'єктам, що відстежуються, стають першими детекціями нових об'єктів. Траєкторії об'єктів, для яких детекція не була знайдена, можуть бути продовжені за допомогою передбачених вікон, що раніше обмежують. Це дозволяє відновити стеження об'єктом після помилок алгоритму детектування (відсутності детекції) [7]. Для зменшення впливу цього способу на неправильні об'єднання треків різних об'єктів, асоціація може бути проведена в кілька етапів з огляду на вік або метод оновлення траєкторії об'єктів на попередніх кадрах. [14].

Стандартним методом для асоціації об'єктів та детекцій є вирішення задачі про призначення [7, 22]. Для вирішення цього завдання застосовуються Угорський алгоритм або побудова дводольного графа для об'єктів і детекцій та знаходження зважених пар сполучень у графі за допомогою алгоритму знаходження потоку мінімальної вартості, а також жадібні методи [22, 23].

1.3.7 Сучасні методи детектування

Отже, з'ясовано, що завдання детектування об'єктів визначається як завдання автоматичної локалізації об'єктів певних класів на зображенні. Метод детектування отримує на вхід зображення, виконує його аналіз і повертає набір обмежуючих вікон (детекцій) та значень ймовірності приналежності класам для всіх локалізованих об'єктів.

Сучасні рішення задачі детектування об'єктів, що показують задовільні метрики якості, використовують нейронні мережі. Розрізняють два види методів детектування: двоетапні та одноетапні. Двоетапні методи поділяють процес локалізації та класифікації об'єктів і показують найвищі значення метрик якості детектування. Одноетапні одночасно передбачають обмежують вікна та значення ступенів належності класам для об'єктів, тому працюють з найбільшою швидкістю. Робота методу детектування представлена на рисунку 1.7 [24].

До методів детектування, що показують найвищі метрики якості, на даний момент можна віднести: Faster R-CNN (двоетапний), Mask R-CNN (двоетапний), YOLOv3 (одноетапний), YOLOv4 (одноетапний), RefineDet (одноетапний), CenterNet (одноетапний) [10].

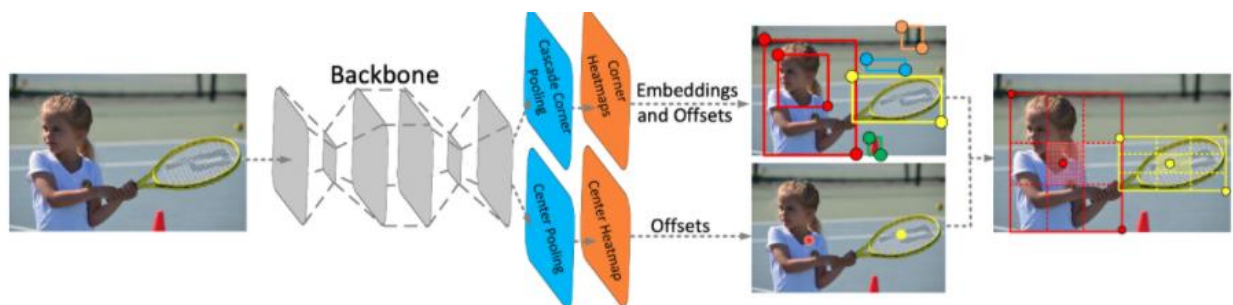


Рисунок 1.7 – Приклад роботи методу детектування об'єктів CenterNet

В алгоритмах трекінгу об'єктів частіше за все використовуються такі одноетапні та двоетапні нейромережні методи детектування як Faster R-CNN [14], SSD [25], детектори серії YOLO [26 – 28].

1.4 Постановка задачі дослідження

Об'єктом дослідження є відео з послідовністю різноракурсних зображень об'єктів.

Метою даної кваліфікаційної роботи є аналіз існуючих методів детекції та методів трекінгу, а також вибір найоптимальніших. Для цього необхідно виконати наступні завдання:

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати існуючі методи детектування об'єктів;
- провести огляд та аналіз існуючих методів вирішення завдання трекінгу;
- розробити власний алгоритм онлайн трекінгу множини об'єктів у режимі реального часу на основі вибраних методів;
- провести дослідження та налаштування алгоритму онлайн трекінгу множини об'єктів за допомогою розроблених програмних засобів.

2 МАТЕМАТИЧНІ МОДЕЛІ ДЕТЕКЦІЇ Й ТРЕКІНГУ ОБ'ЄКТІВ

2.1 Огляд детекції та трекінгу

2.1.1 Задача детекції

Одним із важливих моментів у роботі систем комп'ютерного зору є визначення (детектування) людини на зображенні. Але при спробі побудови автоматичної системи виявлення осіб доводиться стикатися з факторами, що ускладнюють процес детекції:

- індивідуальна особливість рис особи ускладнює вибір ознак об'єктів детекції;
- невелика корекція у положенні голови камери призводить до суттєвої зміни зображення обличчя;
- на зображенні можна побачити лише частину особи, яка прихована іншими предметами;
- умови зйомки (світло, кольорова гама, контрастність та погіршення зображення) впливають на зображення.

Це зумовлено тим, що завдання визначення осіб є необхідність продовжувати навчання нових образах об'єктів. Він називається динамічним навчанням. Основною перевагою систем комп'ютерного зору, які використовують такий вид навчання, є підвищення ефективності детекції в процесі навчання, оскільки з часом алгоритм робить менше помилок під час навчання на нових об'єктах. Після завершення перевірки чергового об'єкта алгоритм передає позначку класу цього об'єкта, тобто вона має бути врахована та «донавчатися».

2.1.2 Математична модель детекції

Нехай є відеопотік, характеризується безперервністю кадрів. Кожен кадр є цифровим зображенням. У пам'яті комп'ютера зображення є

двовимірною матрицею пікселей A_t розміром $m \times n$, в якій кожен піксель представлений деяким невід'ємним числом від 0 до 255 якщо зображення чорно-біле і від 0 до 2553, якщо воно кольорове [29].

Робота з відеопотоком, на відміну від статичних зображень, дозволяє виділити зону області руху, тим самим зменшуючи розмір області пошуку.

Після виділення області руху ми отримаємо матрицю C_t , шляхом видалення її з матриці A_t . Таким чином розмір отриманої матриці виходить $w_c \times h_c$.

З цього, по області руху можемо визначити список прямокутників, що містять обличчя особи:

$$F_i = \{h_i, w_i, x_i, y_i\}, \quad (2.1)$$

де F_i – вектор, що містить координати обличчя;

x_i, y_i – координати центру i -го прямокутника;

w_i – ширина центру i -го прямокутника;

h_i – висота центру i -го прямокутника.

Для пошуку об'єктів використовується «скануюче вікно», що проходить поетапно по зображенню з кроком в 1 піксель з перетворенням масштабу при кожному проході (рис. 2.1). Розмір вікна визначається його висотою H й шириною W .

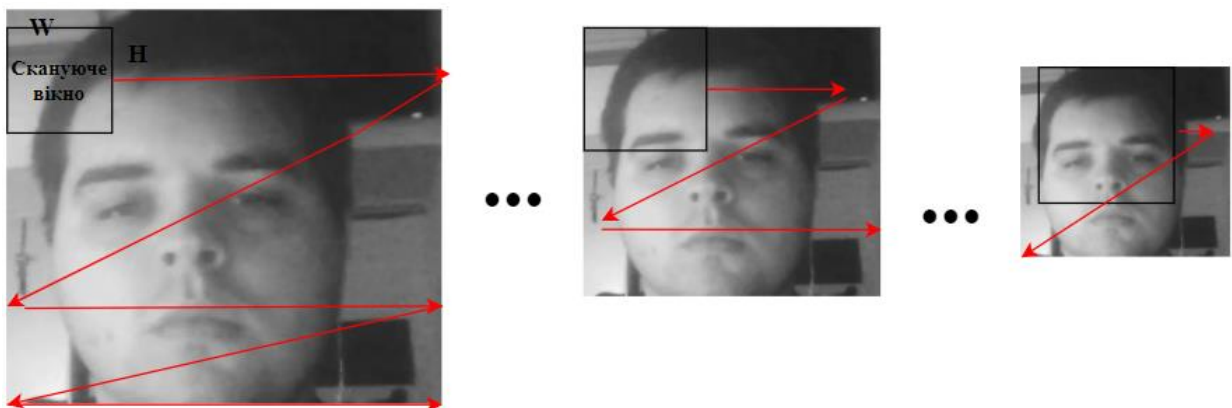


Рисунок 2.1 – Проходження скануючого вікна по зображенню

При переміщенні вікна, що сканує, на зображенні для кожної позиції вікна потрібно визначити, є там обличчя чи ні. Таким чином, можна зробити висновок, що це може робитися шляхом віднесення частини зображення, що вивчається, в один з двох класів. Перший клас – це наявність особи, а другий – без неї. У цьому випадку мета детекції особи полягає до вибору найбільш відповідного класифікатора, який мав би стійкість до перерахованих вище факторів, що ускладнюють його роботу.

Таким чином, з наявністю двох класів, задача детекції виглядає так.

Дана множина X – множина, в якій зберігається опис ознак об'єктів, Y – кінцева множина класів. Функція $F: X \rightarrow Y$ є вирішальною функцією, тобто класифікатор. Признак f об'єкта a – це відображення $f: X \rightarrow D_f$, де D_f – множина припустимих значень f . Якщо задан набір ознак f_1, \dots, f_r для деякого об'єкта a , то вектор ознак x об'єкта $a \in X$ можна визначити таким чином:

$$x = (f_1(a), \dots, f_r(a)). \quad (2.2)$$

При цьому вектор ознак можна ототожнювати з самими об'єктами.

Класифікатор F має бути здатним класифікувати випадковий об'єкт $a \in X$. Оптимальним вважається класифікатор, який видає найменшу імовірність помилки при всіх припустимих значень x .

Для навчання класифікатора F використовується навчальна вибірка, яка задана множиною $D = \{(x_1, y_1), \dots, (x_m, y_m)\}$, де $y_i \in Y$. Загальний алгоритм детекції обличчя в відеопотоці зображено на рисунку 2.2.

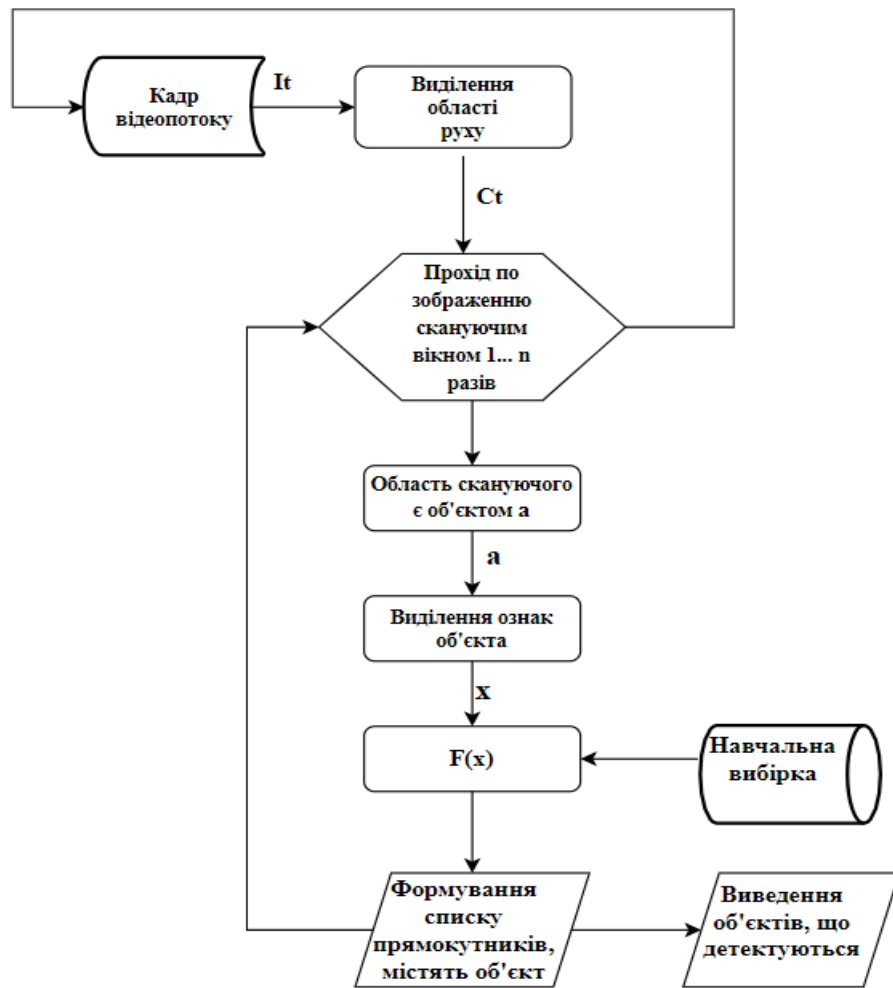


Рисунок 2.2 – Алгоритм детекції обличчя в відеопотоці

2.1.3 Проблематика детекції

Основною проблемою при детекції осіб – це наявність на зображенні, що аналізується великої кількості заважаючої чи несуттєвої шумової інформації.

Зображення, що надходить до класифікатору, може характеризуватися наступними ознаками, що впливають на якість роботи алгоритму:

- кути нахилу та повороти обличчя;
- висота камери та кут нахилу;
- рівень освітлення;
- масштаб обличчя щодо розміру зображення;
- зашумленість зображення.

При вирішенні цієї проблеми пропонується алгоритм двопротинної класифікації, особливість якої полягає у паралельній роботі двох класифікаторів. Алгоритм має стійкість до мінливих умов зйомки та якості кадру [29].

Першим кроком у вирішенні складнішого завдання – ідентифікації (розпізнавання) особи є детекція. При цьому порогові значення повинні бути встановлені таким чином, щоб детектована особа могла бути надалі розпізнана. Враховуючи, що об'єкт детекції має симетрію, для параметрів, що характеризують нахили та повороти обличчя, допустимо брати значення по модулю.

Вплив зовнішніх умов на об'єкт a можна виразити таким чином:

$$Q(a) = (a_h, a_v, a_z, L, N), \quad (2.3)$$

де a_h – боковий нахил обличчя відносно вертикальної осі;

a_v – нахил обличчя відносно вертикальної осі;

a_z – фронтальний кут нахилу голови (залежить від висоти встановлення й кута нахилу камери);

L – рівень освітленості сцени;

N – рівень зашумленості зображення (0 – ідеальне зображення, 1 – нескінченно зіпсоване зображення).

2.1.4 Математична модель трекінгу

Сучасні методи трекінгу рухомих об'єктів, що оперують зміною у зображенні в часі, вимагають великої кількості ресурсів і залежать від багатьох параметрів, наприклад фону (рухомий або нерухомий) та характеру руху (швидкий або повільний). У зв'язку з тим, що в ході розпізнавання особи людини першим етапом завжди йде пошук особи на зображенні (наприклад,

шляхом Віоли Джонса), необхідно використовувати цей процес трекінгу. Знаючи координати зафіксованої особи на зображенні, можна оперувати не масивом пікселя, а лише його відносною зміною розташування в послідовності кадрів. При цьому метод виявлення області обличчя на зображенні досить ресурсозатратний і може займати більше процесорного часу при обробці відеопотоку високої якості на слабких ЕОМ. Такий підхід до вирішення задачі положенні особи на знімку, а й швидку попередню підготовку кадрів відеопотоку з метою знизити область, придатну для використання детектора.

Алгоритми трекінгу, оцінки та компенсації ракурсу зображень осіб рідко розглядаються як компоненти системи розпізнавання, що ускладнює їх адаптацію до розв'язання задачі. Розробка методів, оптимальних для автоматичного застосування в системах реального часу, є критично важливою для формування комплексної технології попереджувальної обробки зображень осіб у відеопотоці.

Відеопотік можна подати у вигляді послідовності кадрів (зображень):

$$V_n = [f_1, f_2, \dots, f_N], \quad (2.4)$$

де N – загальна кількість кадрів;

f_i – поточний кадр.

Будь-яку особу, яка знайдена в кадрі можна описати як набір:

$$F_{C_{ij}} = \{I_{f_{ij}}, R_{f_{ij}}\}, \quad (2.5)$$

де $I_{f_{ij}}$ – зображення особи;

$RI_{f_{ij}}$ – положення зображення особи;

i – номер кадру;

j – номер особи.

При цьому R_f можна представити як набір з двох точок:

$$R_f = \{p_1 = (x_1; y_1), p_2 = (x_2; y_2)\}, \quad (2.6)$$

де p_1 – верхня ліва точка прямокутної області,

p_2 – нижня права точка прямокутної області;

x, y – координати точок.

Таким чином, трек особи k в відеопотоці – це послідовність осіб $F_{C_{ik}}$:

$$T_{rk} = [F_{C_{1k}}, F_{C_{2k}} \dots F_{C_{iNk}}], \quad (2.7)$$

Основна задача у трекінгу полягає у зіставленні положень цільового об'єкта на послідовності кадрів. Таким чином, системи трекінгу зазвичай використовують модель руху, яка описує, як може змінюватися зображення цільового об'єкта при всіляких різних його рухах. У разі стеження за становищем особи передбачається, що об'єкти в наступному кадрі змінюють свою позицію незначно. В зв'язку з цим спробуємо наступний метод трекінгу: Знайдене зображення обличчя належить треку T_{rk} , якщо точка-центр прямокутника Rf , $p_c = \left(\frac{x_1 + x_2}{2}; \frac{y_1 + y_2}{2}\right)$, обличчя $F_{C_{ij}}$ належить області прямокутника Rf обличчя $F_{C_{i-1j}}$, знайденого на попередньому кадрі й який належить до треку T_{rk} (рис. 2.4).

2.2 Методи оцінки якості роботи алгоритму трекінгу множини об'єктів

2.2.1 Методи оцінки якості роботи методу детекції

Для того, щоб обрати для нашої предметної області алгоритм, потрібно його оцінити за допомогою методів оцінки якості.

Перечислимо метрики, які використовуються для оцінки якості роботи методу детектування:

– метрика Class Recall – характеризує здібність методу детектування виявляти об'єкти певного класу. Ця метрика може бути обчислена за формулою:

$$Recall = \frac{TP}{GT}, \quad (2.8)$$

де TP – кількість вірних детекцій певного класу;

GT (Ground truths) – кількість істинних детекцій певного класу.

Чим вище значення Class Recall, тим краще здатність ідентифікувати всі об'єкти певного класу.

– метрика Class Precision – описує частку об'єктів вірно визначених які належать певному класу щодо всіх об'єктів, котрі метод детектування відніс до даного класу. Дана метрика може бути розрахована за формулою:

$$Precision = \frac{TP}{Detections}, \quad (2.9)$$

де TP – кількість вірних детекцій певного класу;

$Detections$ – кількість всіх детекцій для даного класу.

Чим вище значення Class Precision, тим краще здатність відрізнити об'єкти певного класу від інших.

– метрика AP (Average Precision) – визначається як площа області під precision-recall кривою для об'єктів певного класу. Чим вище значення AP, тим

ліпше баланс між Class Precision й Class Recall, а отже й вище ступінь довіри до результатів роботи детектора.

- метрика MAP (Mean Average Precision) – це середнє значення AP для усіх класів.
- метрика AR (Average Recall) – визначається як подвійна площа області під recall-IOU кривою для об'єктів певного класу.
- метрика MAR (Mean Average Recall) – це середнє значення AR для всіх класів.

2.2.2 Методи оцінки якості роботи алгоритму трекінга

Для оцінки якості роботи алгоритму трекінгу множини об'єктів використовуються наступні метрики, що розраховуються на розмічених наборах даних для навчання та тестування алгоритмів трекінгу за допомогою детекцій.

Метрики, що характеризують число правильно чи помилково передбачуваних траєкторій:

- IDsw (ID switches) – загальна кількість випадків, коли траєкторія об'єкту, яка отримана за допомогою алгоритму трекінгу, вірна, але відповідний ідентифікатор помилково змінюється;
- FP (False Positives, False Alarms) – загальна кількість хибних об'єктів;
- FN (False Negatives, Misses) – загальна кількість об'єктів, які не були відстеженні алгоритмом трекінгу;
- MT (Mostly tracked) – загальна кількість об'єктів, для яких було знайдено не менше ніж 80% траєкторії;
- PT (Partially Tracker) – загальна кількість об'єктів, для яких було знайдено 20 - 80% траєкторії;
- ML (Mostly Lost) – загальна кількість об'єктів, для яких було знайдено менше ніж 20% траєкторії;

– Frag (Fragments) – загальна кількість втрат траєкторії об'єктів.

Метрика МОТА (Multiple Object Tracking Accuracy) – дозволяє врахувати всі помилки, які скоює алгоритм трекінгу. Він характеризує здатність трекера підтримувати траєкторії об'єктів, незалежно від їх точності. Для розрахунку МОТА застосовується формула [30]:

$$MOTA = 1 - \frac{(FN+FP+IDsw)}{Ground\ truths} , \quad (2.10)$$

де *Ground truths* – кількість істинних об'єктів.

Чим вище МОТА, тим вище якість роботи алгоритму трекінга. Може приймати й негативні значення з того, що кількість помилок алгоритму може перевищувати кількість істинних об'єктів.

2.3 Розробка архітектури алгоритма трекінгу множини об'єктів

Розроблений алгоритм онлайн трекінгу множини об'єктів у режимі реального часу ґрунтується на підході трекінгу за допомогою детекцій. Робота алгоритму було поділено на блоки, кожен із яких вирішує певне завдання. Методи, які у блоках, розробляються і реалізуються окремо друг від друга. Такий підхід забезпечує можливість дослідження впливу використання різних методів штучного інтелекту на результат роботи алгоритму трекінгу, а також дозволяє гнучке налаштування алгоритму для відповідності різним апаратним вимогам та запитам користувача. Алгоритм працює ітераційно. Блок-схема роботи однієї ітерації алгоритму трекінгу представлена рисунку 2.6.

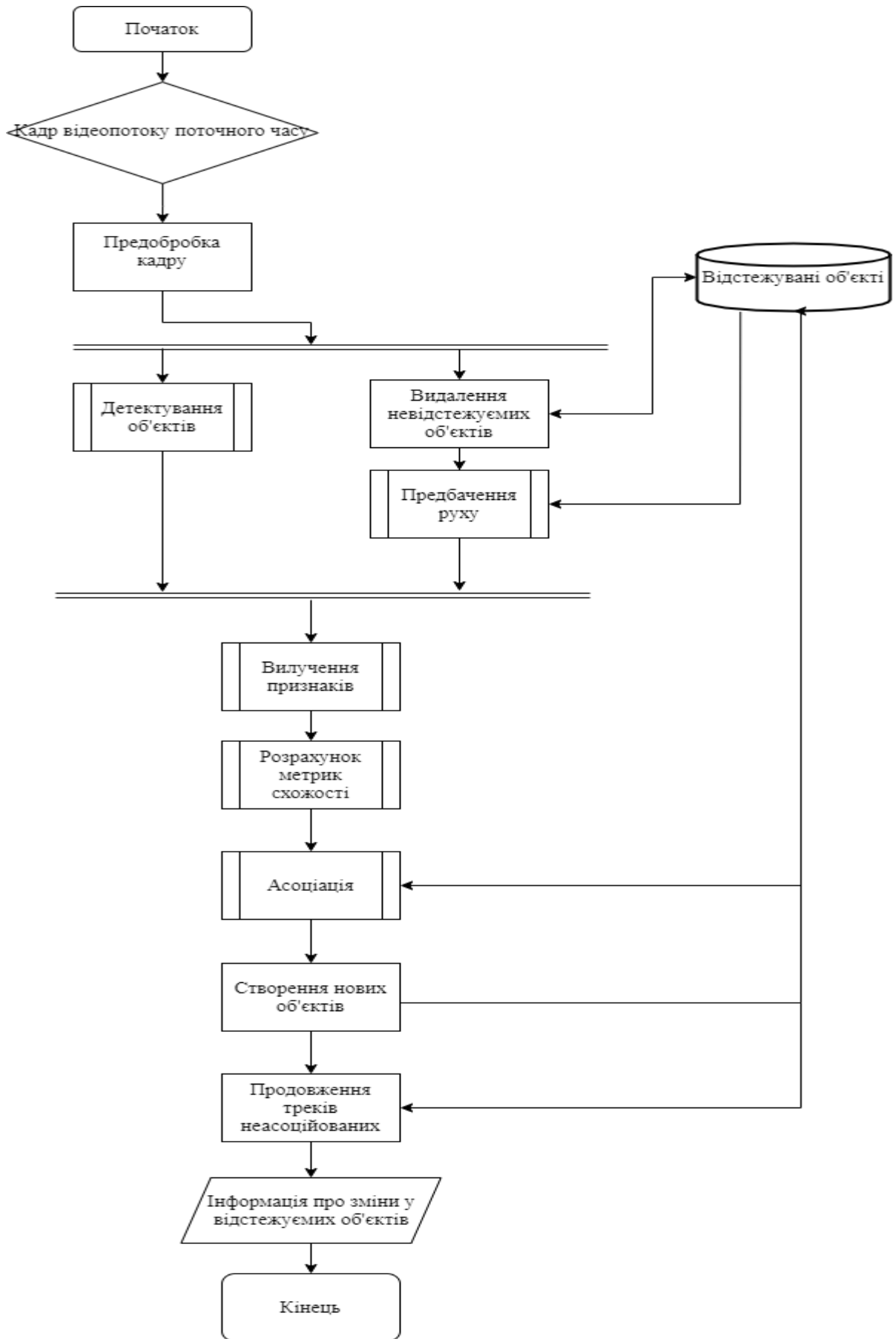


Рисунок 2.6 – Блок-схема алгоритму трекінгу

На кожній ітерації на вхід до алгоритму надходить один кадр відеопослідовності та поточний час (час захоплення кадру). У блоці «Переробка кадру» відбувається підготовка кадру відеопослідовності до використання в алгоритмі трекінгу (масштабування кадру). У блоці «Детектування об'єктів» відбувається локалізація та передбачення значень ймовірностей класів об'єктів, що у результаті формує детекції. Незалежно від цього визначається, які об'єкти, що відстежуються, не оновлювалися довше заздалегідь заданого проміжку часу – такі об'єкти видаляються. Для всіх інших об'єктів, що відстежуються, в блоці «Передбачення руху» розраховуються передбачувані обмежуючі вікна для поточного кадру відеопотоку. Далі для кожного вікна, що обмежує, в блоці «Вилучення ознак» обчислюються вектори графічних ознак. У блоці «Розрахунок метрик схожості» для кожної пари об'єкта, що відстежується, і детекції розраховується зважена сума заздалегідь визначених метрик схожості. У блоці «Асоціація» відбувається зіставлення детекцій об'єктам, що відстежуються. Створюються нові об'єкти для детекцій, які були зіставлені існуючим об'єктам, що відстежуються. Траєкторії об'єктів, для яких не було знайдено відповідних детекцій на етапі асоціації, продовжуються за допомогою обмежувальних вікон, отриманих у блоці «Передбачення руху», деяке заздалегідь задане число кадрів. Інформація про оновлення та створення об'єктів зберігається. Результатом роботи алгоритму трекінгу для кадру відеопотоку є дані про віддалені об'єкти, що обмежують вікна, ймовірність приналежності класам та відповідні ідентифікатори всіх не видалених об'єктів, крім тих, які не були оновлені на поточному кадрі – такі об'єкти зберігаються деякий час для того, їх треки, якщо з'явиться асоційована детекція. Таким чином, життєвий цикл об'єкта може бути описаний наведеною на рисунку 2.7 діаграмою станів. При запиті вихідної інформації алгоритму трекінгу на кожному обробленому кадрі відеопотоку можна отримати повну інформацію про всі об'єкти, які були відстежені алгоритмом на даному відеопотоці.



Рисунок 2.7 – Діаграма станів відстежуваного об'єкта

Далі розглянемо розробку методів для кожного блоку алгоритму трекінгу.

2.4 Розробка методів детектування

Для локалізації об'єктів на кадрах відеопотоку в режимі реального часу було вирішено застосовувати одноетапні нейромережеві методи детектування, що показують високі метрики якості та продуктивності. Виходячи з цього для розробки алгоритму трекінгу були обрані такі методи детектування, що мають реалізацію зі відкритим вихідним кодом:

- метод детектування YOLOv3;
- метод детектування YOLOv4;
- метод детектування CenterNet.

Для всіх алгоритмів детектування були обрані моделі, навчені на наборі даних COCO [31], в якому міститься 330 тисяч зображень, віднесених до 80 класів, у тому числі класів «людина» («person»), «машина», («car»), «велосипед» («bicycle»).

Алгоритми детектування YOLO (You Only Look Once) – популярна архітектура згорткової нейронної мережі для одноетапного вирішення задачі детектування множини на зображення. Швидка швидкість роботи даного алгоритму забезпечується необхідністю застосування операції згортки до зображення лише один раз [32].

У ході роботи алгоритму зображення ділиться на сітку. Для кожного осередку сітки за допомогою нейронної мережі передбачається наступне: певна кількість обмежуючих вікон; ймовірності того, що вікно, що обмежує, є шуканий об'єкт; значення ймовірності того, що об'єкти, центри яких знаходяться в осередку, належать певному класу. Результатом роботи нейронної мережі є набір детекцій. Схема роботи алгоритму детектування YOLO зображено на рисунку 2.8 [32].

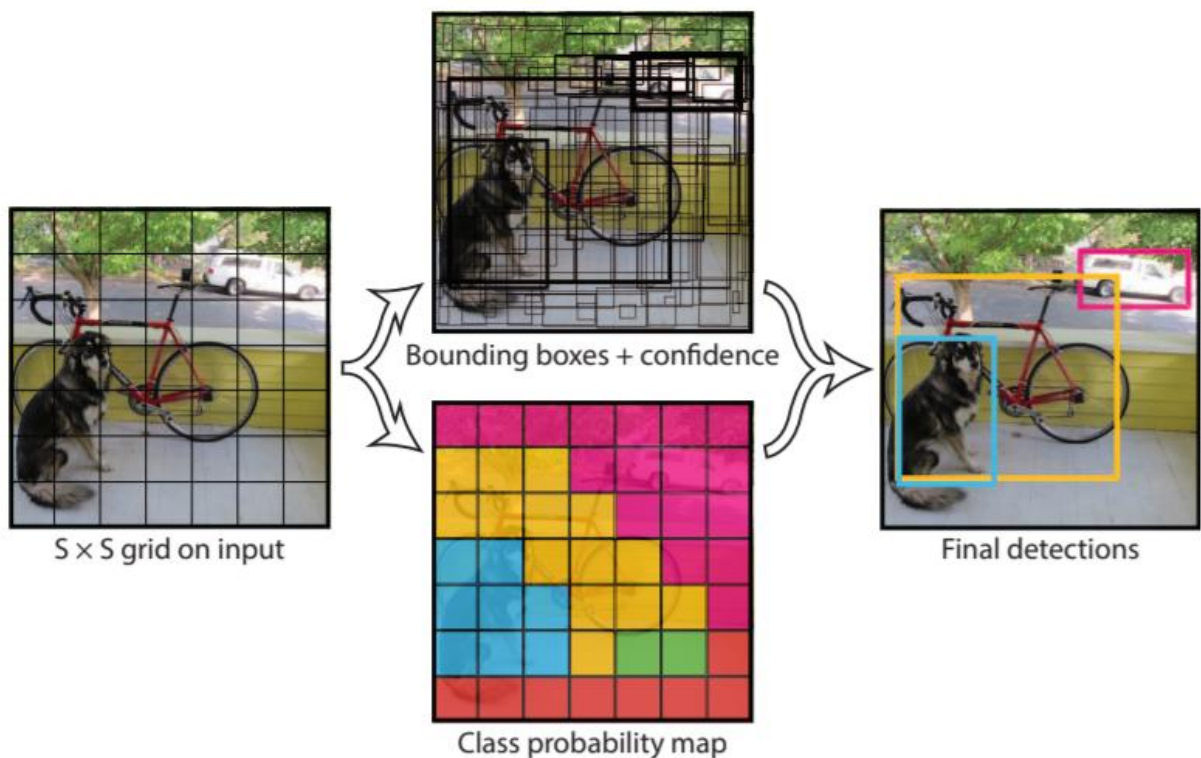


Рисунок 2.8 – Схема роботи детектора об'єктів YOLO

YOLOv3 – одна із останніх версій архітектури YOLO. Порівняно з попередніми версіями, YOLOv3 краще справляється з локалізацією невеликих

об'єктів, при цьому залишаючись одним із найшвидших алгоритмів детектування об'єктів [33].

YOLOv4 – остання версія архітектури YOLO. Ця версія моделі несе в собі ряд модифікацій, спрямованих на пошук оптимального співвідношення якості детектування та продуктивності [34].

CenterNet – архітектура нейронної мережі для одноетапного рішення задачі детектування об'єктів на зображенні, що показала одні з кращих результатів тестів продуктивності при збереженні високих значень метрики якості [35].

CenterNet використовує наступний підхід до нейромережевого детектування об'єктів. Об'єкти представляються у вигляді точок – центрово обмежуючих вікон. Значення ознак у точці описують ймовірність того, що локалізовано шуканий об'єкт, розміри обмежувальних вікон, ймовірності приналежності класам та інші характеристики об'єкта. Модель навчається по кадру визначати набір точок їх ознак, з допомогою яких виходять детекції. Приклад представлення об'єктів як точок показаний на рисунку 2.9 [35].

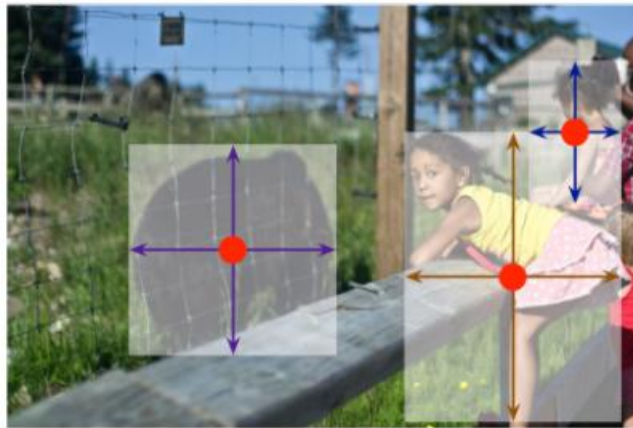


Рисунок 2.9 – Приклад подання об'єктів як точок для алгоритму детектування об'єктів CenterNet

2.5 Розробка методів передбачення руху для алгоритму трекінгу множини об'єктів

Для розробки власного алгоритму трекінгу множини об'єктів у режимі реального часу було відібрано такі методи передбачення руху:

- використання оптичного потоку;
- використання вектора швидкості;
- використання фільтра Калмана;

У розроблюваному алгоритмі трекінгу на етапі передбачення руху була передбачена можливість ініціалізації декількох методів одночасно таким чином, що вікно, що обмежує, розраховується якимось першим методом, який виконується без виникнення винятків і для якого достатньо зібраної інформації, зі списку методів, розташованих у порядку зменшення пріоритету. Додатково було додано можливість відключення передбачення руху шляхом використання останнього обмежуючого вікна об'єкта. Опис розроблених методів передбачення руху наведено далі.

2.5.1 Використання оптичного потоку

Оптичний потік характеризує зміни, що відбулися в сцені, як вектори швидкості для кожного пікселя зображення. Для передбачення руху об'єкта на зображенні за допомогою даних векторів розраховується швидкість об'єкта та, використовуючи інформацію про положення об'єкта на попередньому кадрі, знаходиться нове положення об'єкта. У алгоритмі трекінгу, що розробляється, для розрахунку оптичного потоку використовується алгоритм Фарнебека, який дозволяє визначити пересування частин зображення, використовуючи тільки два послідовні кадри [36]. Даний алгоритм був обраний завдяки своїй продуктивності, порівняно високій якості передбачення руху та необхідності зберігати лише один попередній кадр для розрахунку оптичного потоку щодо поточного кадру відеопослідовності. Приклад результату роботи алгоритму

Фарнебека у вигляді векторів швидкості кожного пікселя зображення, представлені рисунку 2.10.

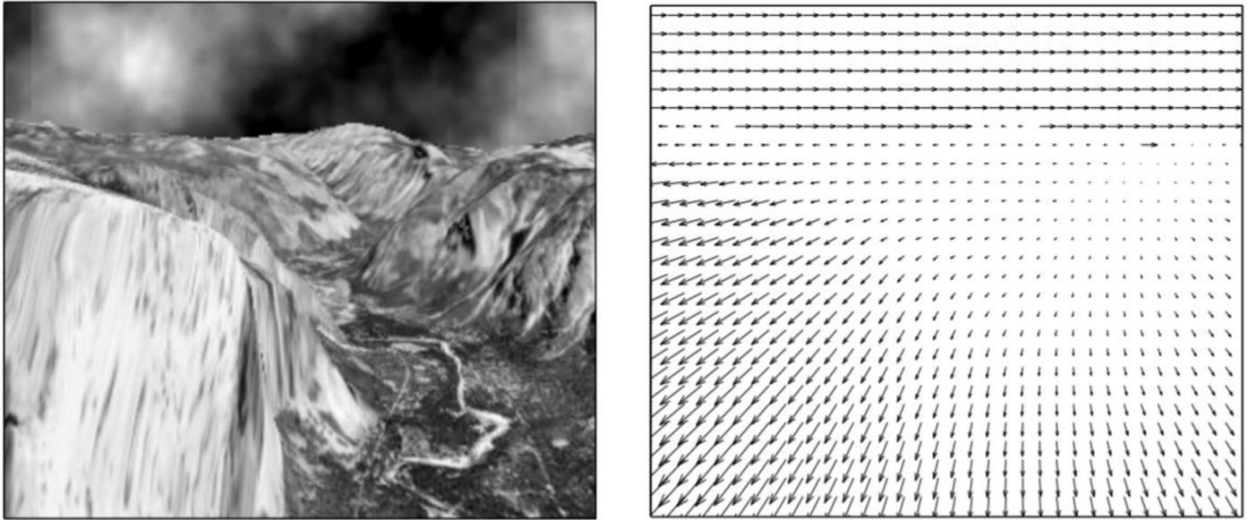


Рисунок 2.10 – Приклад роботи алгоритму Фарнебека

2.5.2 Використання вектора швидкості

Запропоновано алгоритм розрахунку наступного обмеженого вікна об'єкта за допомогою вектора швидкості, який був запропонований раніше: для кожного об'єкта додатково зберігаються часові проміжки між оновленнями траєкторії. За даними, отриманими за допомогою даних, зібраних протягом останньої заздалегідь заданої кількості секунд, для об'єкта, що відстежується, розраховують вектор швидкості на попередньому кадрі. Як було показано раніше, за вектором швидкості об'єкта на попередньому кадрі і тимчасовому проміжку між отриманим і попереднім кадром, визначається поточне місце об'єкта.

2.5.3 Використання фільтра Калмана

Фільтр Калмана дозволяє оцінити стан динамічної системи, використовуючи неповні та неточні виміри. При використанні фільтра Калмана для передбачення руху [37] як динамічна система виступає модель

руху об'єкта, що відстежується, як вимірювання – детекції об'єкта на послідовності кадрів. Для моделювання руху кожного відстежуваного об'єкта за допомогою фільтра Калмана в алгоритмі трекінгу, що розробляється, використовується наступний набір параметрів:

- x координата об'єкта на кадрі;
- y координата об'єкта на кадрі;
- x компонент швидкості об'єкта;
- y компонента швидкості об'єкта;
- висота об'єкта;
- ширина об'єкта;
- швидкість зміни висоти об'єкта;
- швидкість зміни ширини об'єкта.

Таким чином, за допомогою фільтра Калмана моделюється рівномірний рух обмежуючого вікна об'єкта з розміром обмежуючого вікна, що рівномірно змінюється. Експериментальним шляхом було визначено, що при використанні фільтра Калмана як метод прогнозування руху в алгоритмі трекінгу множини об'єктів, найвищі значення метрик якості прогнозування досягаються при використанні як матриць коваріації для випадкового процесу (для моделі руху) і шуму вимірювань (детекцій) діагональні матриці що містять дисперсії кожного параметра вектора стану або виміру на відповідних позиціях. Таким чином, передбачається, що параметри стану та вимірів лінійно незалежні або їх залежності не враховуються. Використання фільтра Калмана на етапі передбачення руху в алгоритмі трекінгу множини об'єктів, що розробляється, відбувається таким чином:

- Для кожного відстежуваного об'єкта при появі його першої детекції створюється екземпляр фільтра Калмана.
- На кожному кадрі для кожного відстежуваного об'єкта за допомогою відповідного фільтра Калмана передбачається положення та розміри об'єкта, що обмежує, на поточному кадрі, які можуть бути використані на етапі асоціації або продовження траєкторій для неасоційованих об'єктів.

– Після завершення етапу асоціації та продовження траєкторій для неасоційованих об'єктів для кожного об'єкта, траєкторія якого була доповнена новим вікном, що обмежує, відбувається корекція відповідного фільтра Калмана за допомогою отриманих вимірювань положення та розмірів об'єкта на поточному кадрі.

На рисунку 2.11 наведено приклад роботи фільтра Калмана. На ньому відображено передбачення п'ятого обмежуючого вікна в траєкторії об'єкта після коригування фільтра Калмана чотирма справжніми вікнами, що обмежують. Зеленим кольором позначаються справжні обмежуючі вікна, червоним – передбачені фільтром Калмана.



Рисунок 2.11 – Приклад роботи фільтра Калмана

Була додана можливість використання стану фільтра Калмана після додавання нового об'єкта, що обмежує, для корекції (фільтрації)

характеристик даного вікна, що обмежує. Таким чином, фільтр Калмана може бути використаний для корекції помилок методу детектування об'єктів.

2.6 Розробка методів вилучення графічних ознак для алгоритму трекінгу множини об'єктів

У алгоритмі трекінгу, що розробляється, використовуються два методи вилучення графічних ознак:

- використання моделі згорткової нейронної мережі MobileNetV2, навченої на наборі даних ImageNet [38], для вилучення графічних ознак з кожного вікна, що обмежує;
- використання згорткових шарів детектора YOLOv4 (мережі darknet) для отримання графічних ознак кадру.

2.7 Розробка методів обчислення метрик схожості для алгоритму трекінгу множини об'єктів

Були відібрано такі методи розрахунку метрик схожості, що застосовуються для розробки власного алгоритму трекінгу:

- IOU (Intersection Over Union);
- косинусна подібність.

При розрахунку підсумкового значення схожості в алгоритмі трекінгу передбачено можливість використання виваженої суми метрик схожості з попередньо заданими вагами. Далі докладніше розглянуті методи обчислення метрик, що використовуються. схожості.

2.7.1 Intersection Over Union

IOU (Intersection Over Union) розраховується як приватна площа перетину двох обмежуючих вікон і площа їх об'єднання. Графічна схема розрахунку IOU представлена на рисунку 2.12 [39].

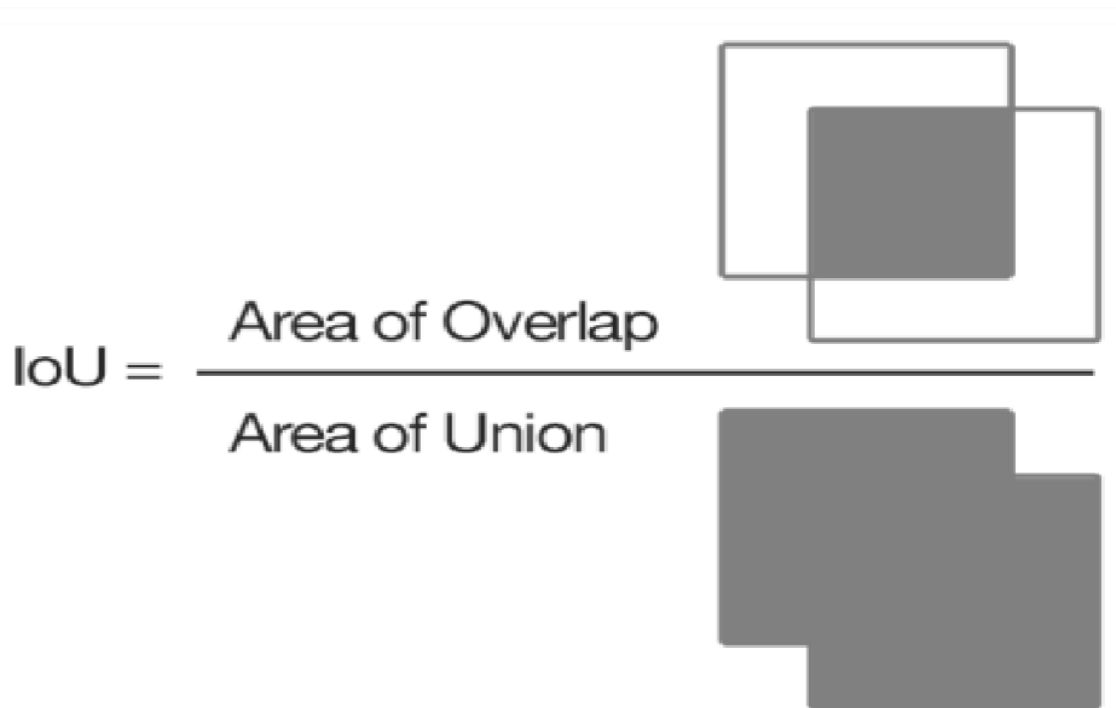


Рисунок 2.12 – Графічна схема розрахунку IOU

Область значень цієї метрики: $[0, 1]$. Чим ближче значення IOU до одиниці, тим більша ймовірність того, що два вікна, що обмежують, належать одному об'єкту.

2.7.2 Косинусна схожість

Косинусна схожість (cosine similarity) характеризує схожість між двома векторами графічних ознак, витягнутих з відповідних вікон, що обмежують. Воно вимірюється косинус кута між векторами і визначає, чи мають дані два

вектори приблизно схожий напрям [40]. Для розрахунку значення метрики косинусної подібності використовується формула:

$$\text{sim}(u, v) = \frac{u \times v}{|u|_2 \times |v|_2},$$

де u, v – вектори.

Область значень цієї метрики: $[0, 1]$. Чим ближче значення косинусної подібності між векторами графічних ознак, що відповідають двом порівнюваним вікнам, що обмежують, до одиниці, тим більша ймовірність того, що ці обмежуючі вікна належать одному об'єкту.

2.8 Розробка методів асоціації

При розробці алгоритму трекінгу, в якому використовуються методи проведення асоціації між об'єктами, що відстежуються (наприклад, за допомогою методу знаходження потоку мінімальної вартості) та детекціями за допомогою алгоритму пошуку мінімуму вартості. Для вирішення задачі про призначення (асоціації об'єктів та детекцій) за допомогою пошукового потоку мінімальної вартості будується дводольний граф таким чином, що вершини однієї частки збігаються з об'єктами, що відстежуються, а вершини другої частки – детекціям. Приклад дводольного графа наведено на рисунку 2.13.

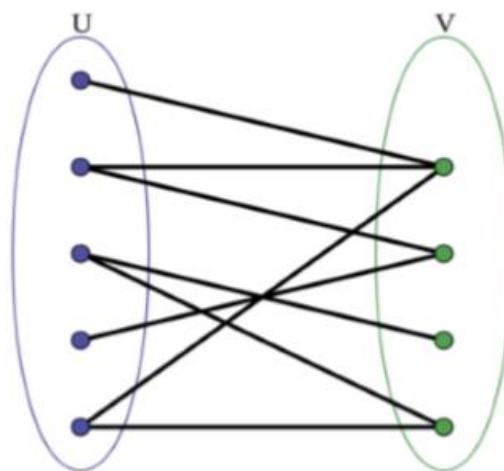


Рисунок 2.13 – Приклад дводольного графа

Як вага (вартості) ребер використовується негативне значення метрики схожості або виваженої суми кількох метрик схожості. Пропускна спроможність кожного ребра становить одну одиницю. Одна частина з'єднується з додатковою вершиною – витокком та двома додатковими вершинами – стоком.

В отриманому графі знаходиться потік мінімальної вартості. З його допомогою знаходиться взаємно однозначна відповідність між вершинами першої частини та вершинами другої частини графа, яка є шуканою асоціацією. [23]. Результати вирішення завдання призначення використовуються зіставлення детекцій і відстежуваних об'єктів.

2.9 Розробка методів оцінки навантаження на основі результатів алгоритма трекінгу множини об'єктів

Для вирішення завдання оцінки трафіку на кадрах відеопослідовності на основі результатів роботи алгоритму трекінгу використовується підхід підрахунку кількості об'єктів певних класів, що перетинають заздалегідь задані для даної відеопослідовності лінії, з розрахунком напряму перетину за певні проміжки часу. Такий підхід був обраний для можливості точно визначити область підрахунку трафіку. Приклад кадру відеопослідовності, на якому позначені лінії підрахунку, а також результати роботи алгоритму трекінгу для даного кадру.

Для зменшення впливу помилок роботи алгоритму трекінгу перед участю в підрахунку об'єкти фільтруються за тривалістю часу життя та максимальним відносним розміром – якщо об'єкт не існував досить довго або займав дуже велику площу кадру, передбачається, що даний об'єкт був породжений хибною детекцією. Таким чином, для підрахунку трафіку за допомогою розробленого алгоритму трекінгу потрібно виконати кроки, представлені на блок-схемі рисунку 2.14.

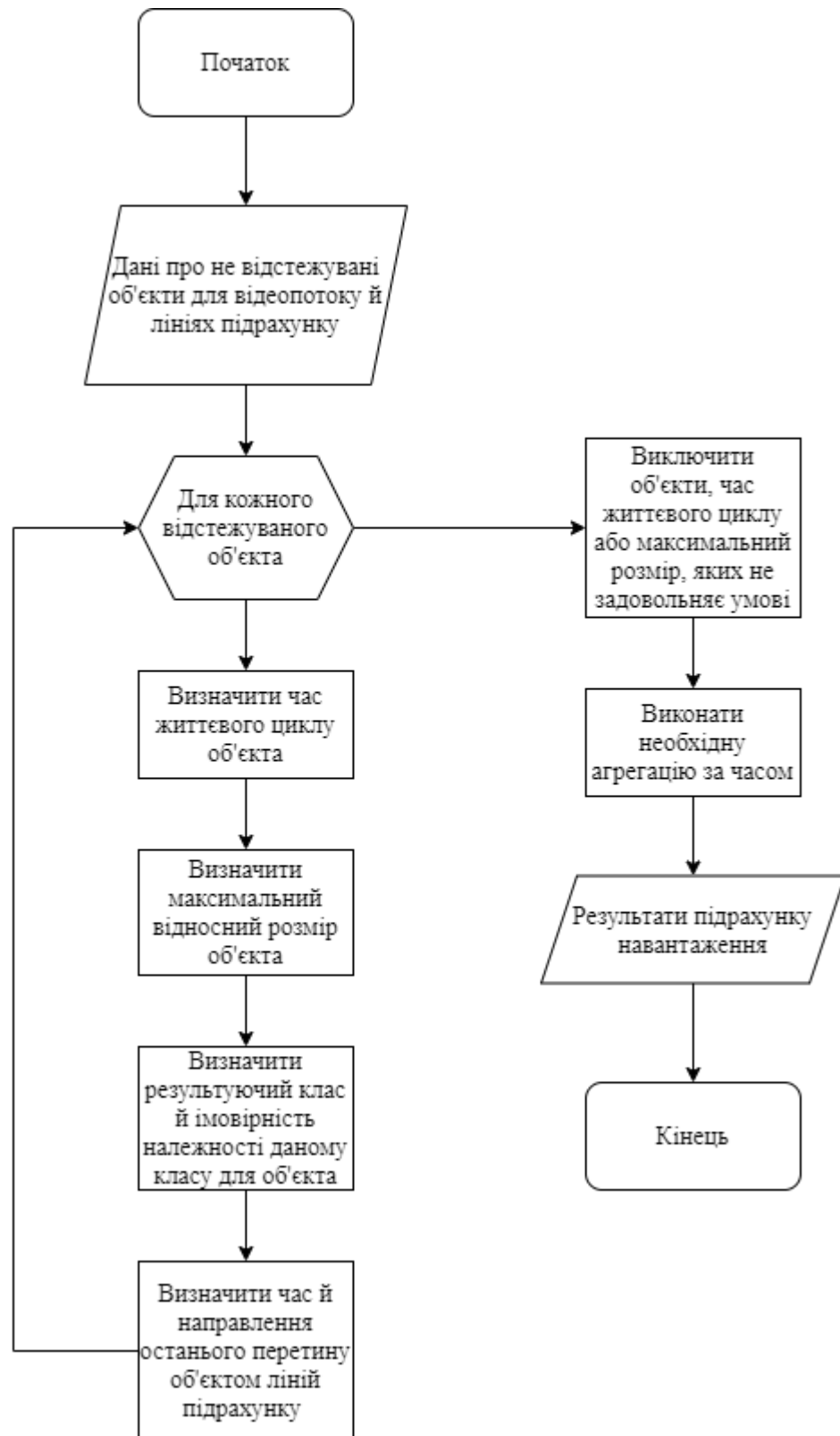


Рисунок 2.14 – Блок-схема алгоритму підрахунку навантаження на основі результатів роботи алгоритму трекінга

Для визначення часу життя об'єкта використовується формула:

$$cycle_time = time.time() - start_time,$$

де $cycle_time$ – час життя об'єкту;

$time.time()$ – час детекції;

$start_time$ – час першої детекції.

Для визначення максимального відносного об'єкта використовується формула:

$$max_size = \max\left(\frac{bbox_size}{frame_size}\right),$$

де max_size – максимальний відносний розмір об'єкту;

$bbox_size$ – площа обмежуючого вікна об'єкта;

$frame_size$ – площа обмежуючого вікна кадру об'єкту.

Для визначення результуючого значення ідентифікатора та ймовірності приналежності класу для об'єкта використовуються середні значення ймовірностей приналежності класам за останнє заздалегідь задана кількість кадрів.

Для визначення останнього часу та напрямку перетину траєкторії з лініями підрахунку перевіряються можливі перетину всіх відрізків траєкторії об'єкта, що з'єднують середини нижніх кордонів вікон, що обмежують. Безпосередньо для визначення факту перетину та розрахунку напрямку перетину використовується метод розрахунку орієнтованої площі [42].

3 КОМП'ЮТЕРНА МОДЕЛЬ ТРЕКІНГА

3.1 Обґрунтування вибору програмної реалізації

У процесі розробки системи автоматичного трекінгу людей використовуються бібліотеки, які дозволяють скоротити обсяг коду, що набирається, а також взаємодіяти з операційною системою користувача.

Для реалізації програмних засобів трекінгу множини об'єктів у режимі реального часу використовуються такі програмні інструменти:

- Мова програмування Python версії 3.7 як основна мова програмування; вибір даної мови пояснюється існуванням безлічі фреймворків і бібліотек для розробки в сферах штучного інтелекту;
- OpenCV для обробки кадрів відеопослідовності, роботи з оптичним потоком і фільтром Калмана;
- NumPy для оптимізації маніпуляцій з даними;
- PyTorch для роботи з моделлю неймережевого детектора CenterNet; для роботи з методами детектування YOLOv3 та YOLOv4.
- Time – бібліотека для роботи з часовими змінними;
- SkLearn для лінійного зіставлення векторів ознак;
- Matplot для візуалізації даних.
- Мови програмування C і C++ для підвищення продуктивності роботи окремих компонентів;

Для роботи з методами детектування YOLOv3 та YOLOv4

Стандартна бібліотека Python не включає інструментів для роботи з комп'ютерним зором. Для цих потреб обрано бібліотеку OpenCV. OpenCV – бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. OpenCV написана на мові високого рівня (C/C++) і містить алгоритми для: інтерпретації зображень, калібрування камери за зразком, усунення оптичних

спотворень, визначення подібності, аналіз переміщення об'єкта, визначення форми об'єкту та стеження за об'єктом. Також бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях вистежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях. Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору.

3.2 Обґрунтування вибору методів детектування для алгоритму

Вибір методу детектування множини об'єктів для відповідного етапу алгоритму трекінгу здійснювався шляхом проведення експериментів для порівняння якості, продуктивності та ресурсоефективності методів. Для етапу детектування алгоритму трекінгу множини об'єктів було реалізовано використання наступних методів (описано у підрозділі 2.4):

- YOLOV3;
- CenterNet;
- YOLOV4.

Проведені експерименти у рамках довідкових випробувань методу детектування, представлені у таблиці 2.1.

Для оцінки якості роботи методу детектування використовують метрики, описані в підрозділі 2.2: Recall, Precision, AP, AR. Значення метрик були отримані на наборі даних COCO, з якого були відібрані зображення, найбільш підходящі для оцінки статичного спостереження. Значення метрик якості методів детектування представлені у таблиці 3.1.

Таблиця 3.1 – Результати розрахунку метрик якості методів детектування

	Recall	Precision	AP	AR
YOLOv3	0,65	0,67	0,46	0,43
CenterNet	0,63	0,66	0,44	0,46
YOLOv4	0,61	0,70	0,44	0,46

Можна помітити, що найкращі показники якості в більшості випадків показує метод YOLOv3, але метод YOLOv4 у деяких випадках показує схожі чи випереджальні значення метрик.

Експерименти були проведені на однаковому наборі кадрів роздільною здатністю 1280 на 720 пікселів. Середня кількість об'єктів на кадрі за секунду становила 19.5 об'єктів. Тривалість тесту для кожного експерименту – 15 хвилин. Результати, отримані в ході експериментів, представлені в таблиці 3.2.

Таблиця 3.2 – Середній час обробки кадра методом детектування

	Середній час обробки кадра, с
YOLOv3	0,041
CenterNet	0,050
YOLOv4	0,024

За результатами проведення експериментів можна відзначити, що метод детектування YOLOv4 значно перевищує інші алгоритми детектування в метриках продуктивності та ресурсоефективності. На сьогоднішній день метод детектування YOLOv4 є одною з останніх версій методу детектування серії YOLO. Даний метод був обраний як фінальний метод детектування при розробці алгоритму трекінгу множини об'єктів, оскільки в ході проведення експериментів було визначено, що серед методів, що показали найвищі метрики якості, YOLOv4 демонструє найвищі метрики продуктивності та ресурсоефективності.

За результатами розрахунку метрик якості можна зробити такі висновки: серед методів передбачення руху, що окремо використовуються, найбільш високі метрики якості трекінгу показує метод розрахунку з використанням фільтра Калмана;

Корекція детекцій за допомогою фільтру Калмана дозволяє значно збільшити значення метрики якості алгоритму трекінгу. Це пов'язано з тим, що існуючі на даний момент методи детектування схильні до помилки під час роботи з реальними даними;

Використання методу вилучення графічних ознак на етапі детектування не покращує значення метрик якості трекінгу, незважаючи на низькі значення помилки передбачення руху при розробці моделі; основною причиною таких результатів є недостатність навчальних даних, яка не дозволяє моделі навчитися однаково добре працювати в різних умовах – на тестових даних та у складі алгоритму трекінгу;

Найвищі метрики якості алгоритму трекінгу досягаються при використанні на етапі передбачення руху комбінації з розрахунків за допомогою фільтра Калмана, по зсуву швидкості, з останнього обмежуючого вікна, а також при додатковому використанні графічних ознак, вилучених з подібності .

Таким чином, в результаті проведення експериментів у рамках дослідження роботи алгоритму трекінгу було визначено наступне:

- позитивний вплив на якість роботи алгоритму трекінгу надають методи передбачення, здатні справлятися з різнорідними даними та коригувати помилки методів детектування;

- позитивний вплив на якість роботи алгоритму трекінгу надає використання графічних ознак, витягнутих з об'єктів, що обмежують вікна, для розрахунку метрик схожості;

- при подальшій розробці методів штучного інтелекту для застосування на різних етапах роботи алгоритму трекінгу варто приділити увагу підготовці навчальних даних для використовуваних моделей нейронних


мереж з метою підвищення їх узагальнюючої здатності, а також оптимізації використання даних методів у складі алгоритму трекінгу (наприклад, використання методів паралельного програмування) при вилучення графічних ознак з вікон, що обмежують.

3.3 Тестування розробленої системи





Через відсутність умов приближених до реальних тестування системи трекінгу проводилась на наборі даних MOT16 [40], котрий містить 14 наборів відеоряду, що складається з зображень. Відстеження й оцінка виконується в координатах зображення. Всі послідовності виконані в координатах, анотовані з високою точністю.

В таблиці 3.3 наведено краткий опис тестувальних даних.

Таблиця 3.3 – Краткий опис тестувальних даних

Приклад сцени	Назва	Роздільна здатність	Кількість рамок виявлень
	MOT16-01	1920 x 1080	6395

Продовження таблиці 3.3

	MOT16-02	1920 x 1080	17883
	MOT16-03	1920 x 1080	10456
	MOT16-04	1920 x 1080	47557
	MOT16-13	1920 x 1080	11450

Продовження таблиці 3.3

	MOT16-14	1920 x 1080	18483
---	----------	-------------	-------

3.4 Ілюстрація роботи системи

Розроблена система дозволяє загрузити відео файл, а також має здібність працювати в режимі реального часу, використовуючи зовнішній пристрій який підключено до порту (камеру).

Приклад роботи програми з відеофайлами тестувальних даних, які наведені у таблиці 3.2 зображено на рисунках 3.1 – 3.4.

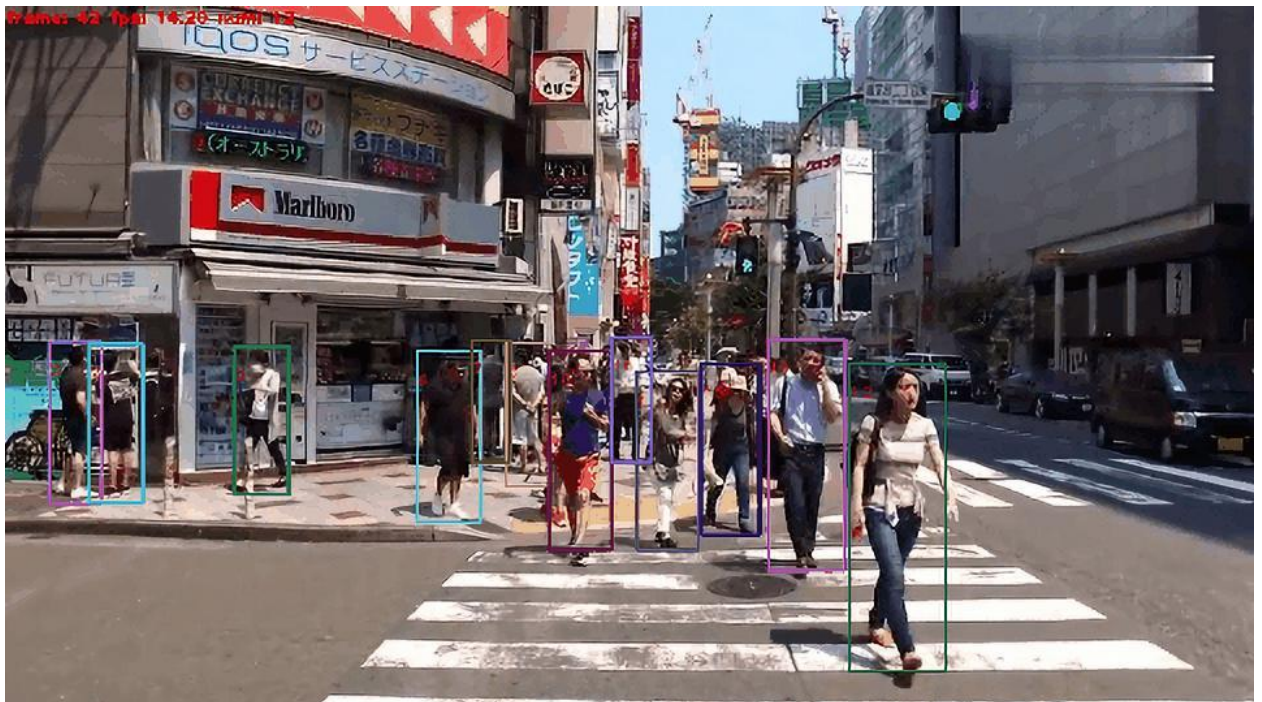


Рисунок 3.1 – Приклад роботи програми з відеофайлом

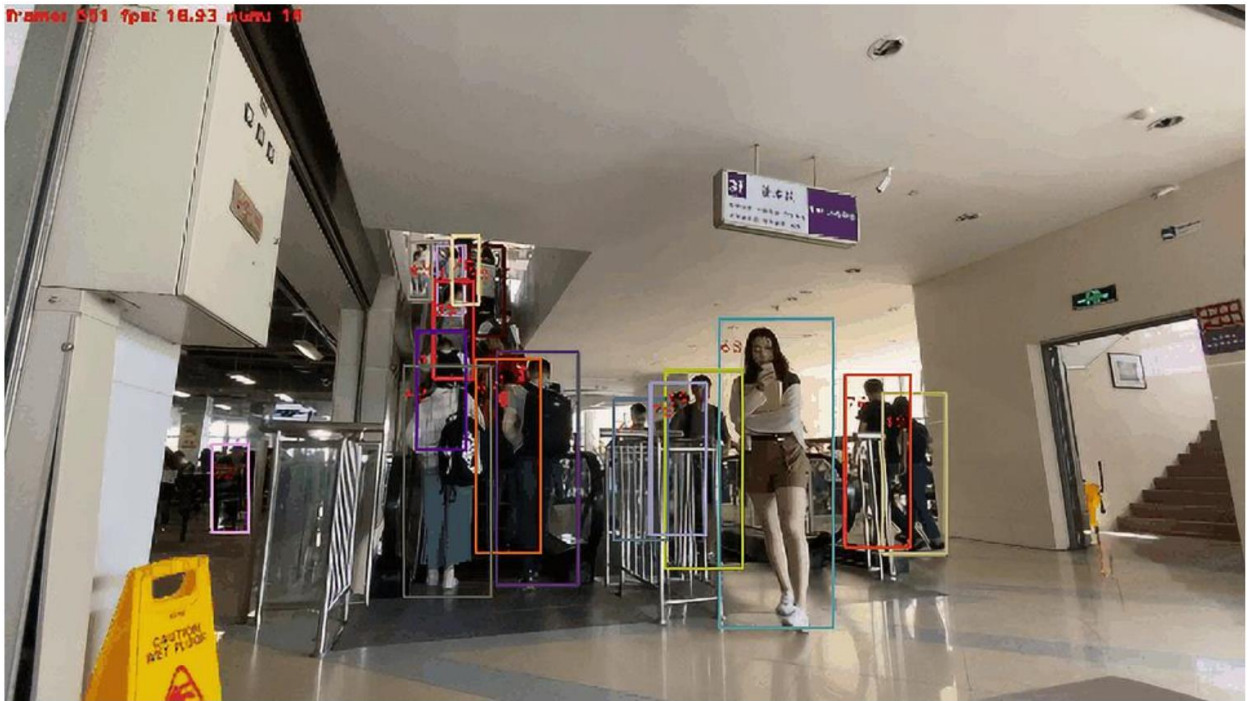


Рисунок 3.2 – Приклад роботи програми з відеофайлом

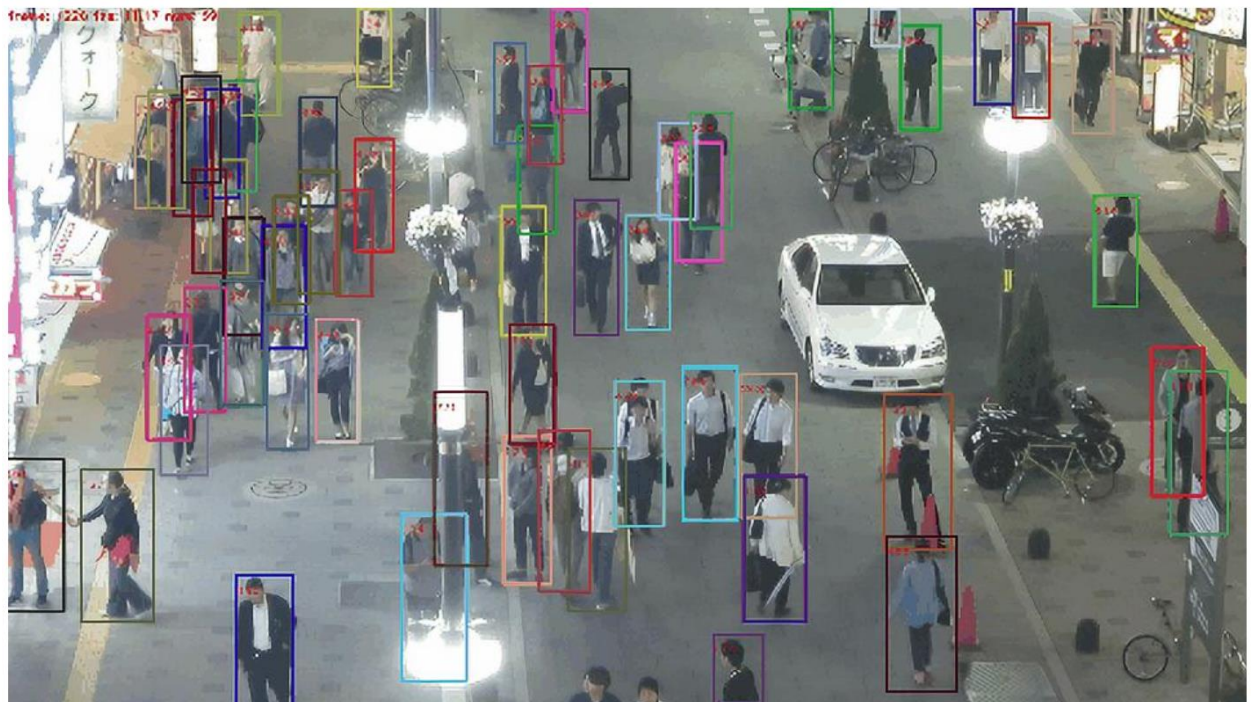


Рисунок 3.3 – Приклад роботи програми з відеофайлом

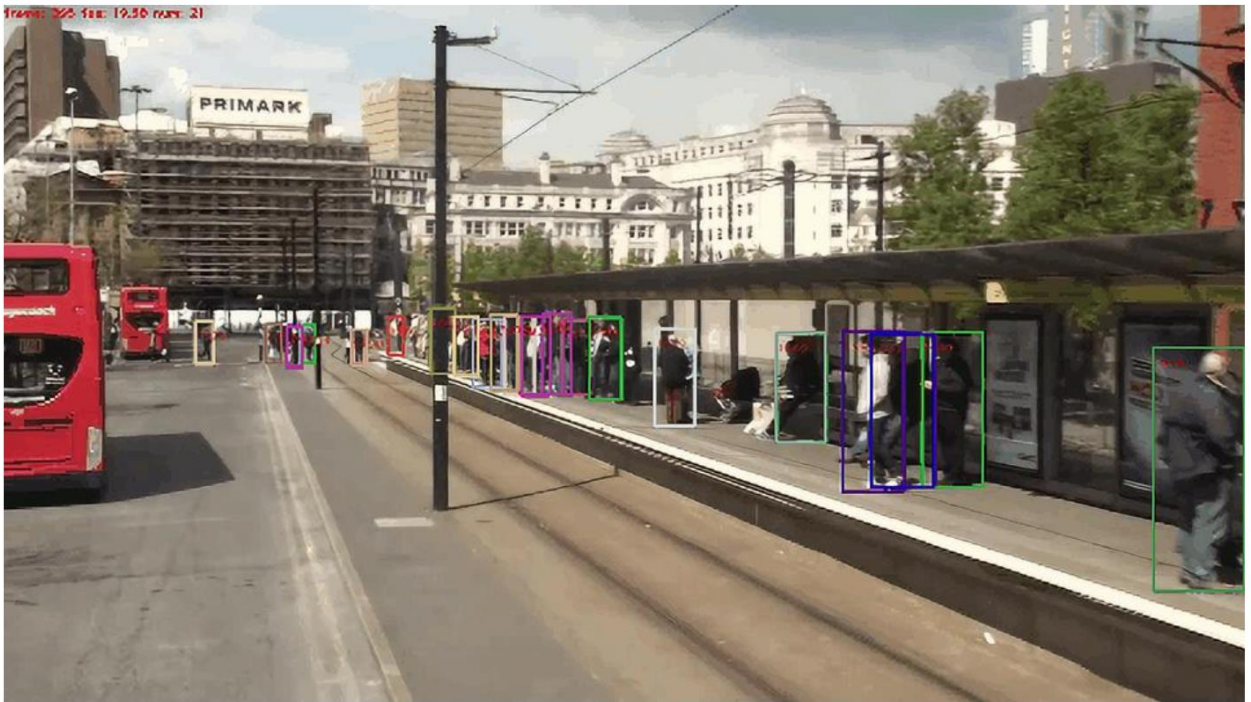


Рисунок 3.4 – Приклад роботи програми з відеофайлом

Робота програми в режимі реального часу зображено на рисунку 3.5.

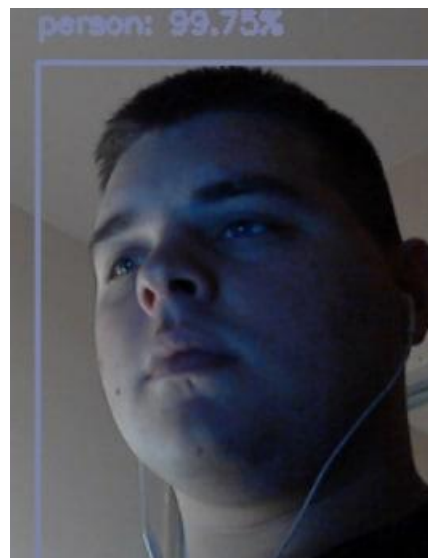


Рисунок 3.5 – Робота програми в режимі реального часу з одним об'єктом

```
Terminal: Local x + v
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
Object detected: person: 99.99%
(300, 400, 3)
[INFO] Elapsed Time: 22863.86
[INFO] Approximate FPS: 10.00
```

Рисунок 3.6 –Результат роботи програми

Робота програми з даними, які наближені до реальних зображено на рисунку 3.7.



Рисунок 3.7 – Робота програми в умовах наближених до реальних з множиною об'єктів

```
(300, 400, 3)
Object detected: person: 85.66%
Object detected: person: 51.65%
Object detected: person: 43.65%
```

Рисунок 3.8 – Результат роботи програми з множиною об'єктів

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований метод трекінгу в момент реального часу .

У ході виконання роботи було отримано такі результати:

- вивчено існуючі методи, що застосовуються при вирішенні завдання трекінгу;

- особливістю розробленого алгоритму є можливість гнучкого налаштування застосовуваних різних етапах роботи методів штучного інтелекту;

- дослідження застосування детекції на різних етапах роботи алгоритму трекінгу множини об'єктів показали, що в подальшій розробці слід приділяти увагу здатності методів працювати в різних умовах із зашумленими даними, а також оптимізації використання методів у складі алгоритму трекінгу;

- в результаті проведення експериментів було обрано конфігурацію методів штучного інтелекту для застосування на різних етапах онлайн алгоритму трекінгу в режимі реального часу (МОТА 0,3808, обробка до 40 кадрів в секунду. Такі методи, як нейромережевий метод детектування YOLOv4 для локалізації та класифікації об'єктів, використання фільтра Калмана та вектора швидкості для передбачення руху, використання IOU в якості метрики схожості та алгоритму пошуку потоку мінімальної вартості для асоціації детекцій та відстежуваних об'єктів;

- розроблено програмне забезпечення на основі обраних методів.

Результати роботи апробовано у вигляді 2 тез доповідей під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ» [3], Міжнародної науково-технічної конференції «Інформатика, управління й штучний інтелект (ІУШІ – 2021) [6].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Y. Taigman, M. Yang, Marc'Aurelio Ranzato, Lior Wolf. *DeepFace: "Closing the Gap to Human Level Performance in Face Verification"* IEEE Conference on 2014, Computer Vision and Pattern Recognition – Columbus, OH, USA.
2. Аналіз існуючих підходів до розпізнавання
URL: <https://habr.com/ru/company/synesis/blog/238129> (дата звернення 25.10.2021)
3. Lubchenko, V., & Podvalnyi, Y. (2021). Finding and classification of objects.
4. Ciarrone, G., Sánchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R., & Herrera, F. (2020). Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381, pp. 61-88.
5. Зайцева, Е. В., & Степанова, А. Л. (2013). Классификация современных методов трекинга объектов в интеллектуальных системах видеонаблюдения. *Горный информационно-аналитический бюллетень (научно-технический журнал)*, (5).
6. Lubchenko, V., & Podvalnyi, Y. (2021). Tools for following (tracking) objects.
7. Schulter, S., Vernaza, P., Choi, W., & Chandraker, M. (2017). Deep network flow for multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* - construct an end-to-end deep learning min-cost network flow, pp. 2730–2739.
8. Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., & Yu, N. (2017). Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE international conference on computer vision*, pp. 4836-4845.

9. Jiang, M. X., Deng, C., Pan, Z. G., Wang, L. F., & Sun, X. (2018). Multiobject tracking in videos based on lstm and deep reinforcement learning. *Complexity*, 2018.
10. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE access*, 7, 128837-128868.
11. Bullinger, S., Bodensteiner, C., & Arens, M. (2017, September). Instance flow based online multiple object tracking. In *2017 IEEE International Conference on Image Processing (ICIP)* IEEE, pp. 785-789.
12. Sheng, H., Zhang, Y., Chen, J., Xiong, Z., & Zhang, J. (2018). Heterogeneous association graph fusion for target association in multiple object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11), pp. 3269-3280.
13. Chen, J., Sheng, H., Zhang, Y., & Xiong, Z. (2017). Enhancing detection model for multiple hypothesis tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 18-27.
14. Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, pp. 3645-3649.
15. Kim, S. J., Nam, J. Y., & Ko, B. C. (2018). Online tracker optimization for multi-pedestrian tracking using a moving vehicle camera. *IEEE Access*, 6, pp. 48675-48687.
16. Fang, K., Xiang, Y., Li, X., & Savarese, S. (2018, March). Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 466-475.
17. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), pp. 1137-1149.
18. Ren, S., He, K., GirshicN, R., & Sun, J. (2017). 9Faster RYCINN: towards RealY tiMe Object Detection with Region proPosal NetworNs, 9 in IEEE

transactions on pattern analysis and machine Intelligence, vol. 39, no. 6. *PP. 1137Y1149, 1.*

19. Lu, Y., Lu, C., & Tang, C. K. (2017). Online video object detection using association LSTM. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2344-2352.

20. Sharma, S., Ansari, J. A., Murthy, J. K., & Krishna, K. M. (2018, May). Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3508-3515.

21. Ma, C., Yang, C., Yang, F., Zhuang, Y., Zhang, Z., Jia, H., & Xie, X. (2018, July). Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, pp. 1-6.

22. Венгерский алгоритм решения задачи о назначениях. 2012.

URL: https://e-maxx.ru/algorithm/assignment_hungary (дата звернения 16.11.2021).

23. Задача о назначениях. Решение с помощью min-cost-flow . 2011.

URL: https://e-maxx.ru/algorithm/assignment_mincostflow (дата звернения 16.11.2021).

24. Новые архитектуры нейросетей.

URL:<https://habr.com/ru/post/498168/#CenterNet> (дата звернения: 24.10.2021).

25. Kieritz, H., Hubner, W., & Arens, M. (2018). Joint detection and online multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1459-1467.

26. Минаев, Е. Ю., Кутикова, В. В., & Никоноров, А. В. (2018). Трекинг объектов в видеопотоке на основе сверточных нейронных сетей и фрактального анализа. In *Сборник трудов IV международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2018)*, pp. 2792-2798.

27. Alvar, S. R., & Bajić, I. V. (2018, August). MV-YOLO: Motion vector-aided tracking by semantic object detection. In *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, pp. 1-5.

28. Hossain, S., & Lee, D. J. (2019). Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices. *Sensors*, *19*(15), 3371.

29. Алгоритми детекції в відеопотоці, 2018.

URL:https://masters.donntu.org/2018/fknt/konoshenko/library/modifitsirovannyu-algorithm-deteksii-lits-v-videopotoke-i-ego-programmnaya_realizatsiya.pdf
(дата звернення 12.11.2021)

30. Bernardin, K., Elbs, A., & Stiefelhagen, R. (2006, May). Multiple object tracking performance metrics and evaluation in a smart room environment. In *Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV* (Vol. 90, No. 91). Citeseer.

31. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, Cham, pp. 740-755.

32. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788.

33. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

34. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

35. Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. *arXiv preprint arXiv:1904.07850*.

36. Farnebäck, G. (2003, June). Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*. Springer, Berlin, Heidelberg, pp. 363-370.

37. Xiang, J., Zhang, G., & Hou, J. (2019). Online multi-object tracking based on feature representation and bayesian filtering within a deep learning architecture. *IEEE Access*, 7, pp. 27923-27935.

38. ImageNet. 2016.

URL: <http://www.image-net.org/> (дата звернення: 17.10.2021).

39. Rosebrock A. Intersection over Union (IoU) for object detection. 2016.

URL: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-objectdetection> (дата звернення 17.10.2021).

40. Hosang, J., Benenson, R., Dollár, P., & Schiele, B. (2015). What makes for effective detection proposals?. *IEEE transactions on pattern analysis and machine intelligence*, 38(4), pp. 814-830.

41. Проверка двух отрезков на пересечение. 2011.

URL: https://e-maxx.ru/algo/oriented_area (дата звернення 18.11.2021).

42. MOT16: A Benchmark for Multiple Object Tracking

URL: <https://motchallenge.net/data/MOT16> (дата звернення 20.11.2021)