

Вінницький національний технічний університет
Факультет комп'ютерних систем та автоматики
Кафедра Автоматизації та інтелектуальних інформаційних технологій

Пояснювальна записка
до магістерської кваліфікаційної роботи

на тему: «Розробка програмних засобів для системи відстеження рухомих об'єктів у
відео»

Виконав: студент 2 курсу, групи 1АКІТ-18м
спеціальності
151 – «Автоматизація та комп'ютерно-інтегровані
технології»

(шифр і назва напрямку підготовки, спеціальності)

Бондарчук А. С.

(прізвище та ініціали)

Керівник к.т.н., доц. Маслій Р. В.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Вінниця ВНТУ 2019 року

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет комп'ютерних систем та автоматики

Кафедра автоматизації та інтелектуальних інформаційних технологій

Освітньо-кваліфікаційний рівень магістр _____

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

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

Р.Н. Кветний

З А В Д А Н Н Я

На магістерську кваліфікаційну роботу студенту

Бондарчуку Андрію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка програмних засобів для системи відстеження рухомих об'єктів у відео

керівник проекту (роботи) к.т.н., доц. Маслій Р.В

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від «__» _____ 2019 р. № _____

2. Строк подання студентом проекту (роботи) _____

3. Вихідні дані до проекту (роботи) _____

Технології визначення та відстеження об'єктів у відео. Частота кадрів не більше 60 Гц. Нерухома камера. Швидкість руху об'єкту не більше 100 пікселів на кадр. Характер руху: 1) поступальний по прямолінійній або криволінійній траєкторії; 2) періодичний (коливальний або по замкнутій траєкторії). Мова програмування C++. Передбачення положення об'єкту за допомогою фільтра Калмана.

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити): проаналізувати існуючі рішення для визначення та супроводження об'єктів у відео; порівняти платформи розробки програми; розробити структуру програми; реалізувати програму супроводження об'єктів в відео послідовності;

протестувати програму.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів тощо): блок-схема алгоритму роботи – плакат. Порівняння технологій відстеження об'єктів – плакат. Результати тестування програми – плакат.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Маслій Р.В., к.т.н., доц. каф. АІТ		
2	Маслій Р.В., к.т.н., доц. каф. АІТ		
3	Маслій Р.В., к.т.н., доц. каф. АІТ		
4	Маслій Р.В., к.т.н., доц. каф. АІТ		
5	Козловський В.О., к.т.н., проф. каф. ЕПВМ		

7. Дата видачі завдання _____ 2019 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання		
2	Збір інформації		
3	Аналіз існуючих рішень		
4	Вибір інструментів розробки		
5	Розробка алгоритму рішення задачі		
6	Розробка програми супроводження об'єктів у відео послідовності		
7	Тестування програми		
8	Оформлення дипломної роботи		
9	Отримання допуску до захисту		

Студент _____
(підпис)

Бондарчук А.С.
(прізвище та ініціали)

Керівник проекту (роботи) _____
(підпис)

Маслій Р. В.
(прізвище та ініціали)

ANNOTATION

Multiple object tracking consists of detecting and identifying objects in video. In some applications, such as robotics and surveillance, it is desired that the tracking is performed in real-time. This poses a challenge in that it requires the algorithm to run as fast as the frame-rate of the video. Today's top performing tracking methods run at only a few frames per second, and can thus not be used in real-time. I propose that one way of running a method in real-time is to not look at every frame, but skip frames to make the video have the same frame-rate as the tracking method.

In this project, I implement a multiple object tracker, following the *tracking-by-detection* paradigm, as an extension of an existing method. It works by modelling the movement of objects by solving the filtering problem, and associating detections with predicted new locations in new frames using the Hungarian algorithm. Three different similarity measures are used, which use the location and shape of the bounding boxes. Compared to other trackers on the MOTChallenge leaderboard, our method, referred to as *C++SORT*, is the fastest submission, while also achieving decent score on other metrics. By running our model on the dataset, sampled at different frame-rates, we show that the performance is greatly reduced when running the model including detecting objects in real time. In most metrics, the score is reduced by 50%, but in certain cases as much as 90%. This indicates that others, trusting methods, can't be used for real-time tracking, but that needs to be specifically investigated.

АНОТАЦІЯ

Відстеження декількох об'єктів складається з виявлення та ідентифікації об'єктів у відео. У деяких додатках, таких як робототехніка та спостереження, бажано, щоб відстеження проводилося в режимі реального часу. Це створює проблему в тому, щоб алгоритм працював так само швидко, як частота кадрів відео. Сьогодні найбільш ефективні методи відстеження працюють лише з декількома кадрами в секунду, тому їх не можна використовувати в режимі реального часу. Один із способів запуску методу в режимі реального часу - це не дивитись на кожен кадр, а пропускати кадри, щоб відео отримало таку ж частоту кадрів, як метод відстеження. Однак ми віримо, що це призведе до зниження продуктивності.

У цьому проекті я реалізую декілька об'єктних трекерів, слідуючи парадигмі відстеження за виявленням, як розширення існуючого методу. Він працює за допомогою моделювання руху об'єктів шляхом вирішення проблеми фільтрації та асоціювання виявлення з передбачуваними новими місцями в нових кадрах за допомогою угорського алгоритму. Використовуються три різні подібності, які використовують розташування та форму обмежувальних боксів. Порівняно з іншими трекерами в таблиці MOTChallenge, наш метод, який називається C++ SORT, є найшвидшим поданням, а також досягає гідних балів за іншими показниками. Запускаючи нашу модель на наборі даних вибірки з різною швидкістю кадрів, ми показуємо, що продуктивність значно знижується при запуску моделі включаючи виявлення об'єктів у режимі реального часу. У більшості показників оцінка знижується на 50%, але в деяких випадках на 90%. Це вказує на те, що інші, повільніші методи не можуть бути використані для відстеження в режимі реального часу, але для цього потрібно конкретніше дослідження.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	11
1.1 Загальні відомості	11
1.2 Особливості і проблематика.....	12
1.3 Способи зображення об'єктів	13
1.4 Аналіз систем відеоспостереження	15
1.5 Відеокамери, системи відеоспостереження.....	17
1.6 Технічні характеристики відеокамер	19
1.7 Спеціальна відеоапаратура.....	21
1.8 Визначення проблеми	25
1.9 Дослідницьке питання	29
1.10 Обмеження	29
2 ВІДСТЕЖЕННЯ ДЕКІЛЬКОХ ОБ'ЄКТІВ	31
2.1 Відстеження багатьох об'єктів	31
2.2 Відстежування за допомогою виявлення.....	32
2.2.1 Виявлення.....	34
3 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАЮЕЗПЕЧЕННЯ.....	36
3.1 C++SORT.....	36
3.1.1 Виявлення.....	37
3.1.2 Алгоритм передбачення	38
3.1.3 Об'єднання	40
3.2 MOTChallenge.....	42
3.3 Відстежування у різних кадрах.....	42
3.3.1 Виявлення та відстежування об'єктів у реальному часі	43
3.4 Програмне середовище	44
4 РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ	46
4.1 MOTChallenge.....	46
4.2 Відстеження з різною частотою кадрів.....	50
4.2.1 Відстеження в реальному часі	52
5 ЕКОНОМІЧНИЙ РОЗДІЛ	54
5.1 Технологічний аудит розробленої системи відеоспостереження на	54
базі рухомого об'єкта.....	54

5.2 Розрахунок витрат на проведення досліджень.....	58
5.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки	63
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	71
Додаток А (обов'язковий) Технічне завдання.....	79
ДОДАТОК В (ОБОВ'ЯЗКОВИЙ) ЛІСТИНГ ПРОГРАМИ	88

ВСТУП

Актуальність теми. Сучасні системи відеоспостереження відіграють важливу роль у підвищенні рівня безпеки та дотримання прав приватної власності об'єкта під охороною. Щодня компанії, що працюють у сфері систем охорони, розробляють нові технології, які здатні ефективно усувати несприятливі ситуації та підвищувати рівень безпеки. Незважаючи на те, що традиційні системи відеоспостереження є досить поширеними в сучасних охоронних системах та демонструють високий рівень захисту, вони мають певні недоліки. Основним недоліком таких систем є високі затрати пам'яті на збереження зафіксованого відеоматеріалу, оскільки подібні системи постійно фіксують ситуацію на закріпленій території та зберігають велику кількість відеоматеріалу, навіть коли в місці спостереження нічого не відбувається. Це також приводить до збільшення часу, що витрачається на перегляд та аналіз збережених матеріалів. Тому, щоб підвищити ефективність систем відеоспостереження, потрібно позбутися наведеного недоліку, що і стало предметом даного дослідження.

Об'єктом дослідження є система відеоспостереження, а також розпізнавання руху об'єкта в системі відеоспостереження.

Предметом дослідження є програмні засоби для підвищення ефективності систем відеоспостереження.

Мета роботи: зменшення розміру пам'яті, затраченої на збереження відеоматеріалів, а також розробка програмних засобів розпізнавання руху об'єкта для інтеграції у системи відеоспостереження.

Наукова новизна отриманих результатів.

Запропоновано новий метод відстеження об'єктів у відео, який на відміну від існуючих використовує нейромережеву модель уоло для відстеження об'єктів та фільтр Калмана для передбачення місцезнаходження об'єктів у нових кадрах що дозволяє підвищити вірогідність відстеження при короткочасних оклюзіях.

Практична цінність отриманих в роботі результатів полягає в тому, що запропонована система надає можливість значно зменшити розмір пам'яті, відведений для збереження відеоматеріалів. Крім того, запропонована система може бути інтегрована в сучасні системи відеоспостереження майже без апаратних змін.

Апробація і публікація результатів роботи. Результати роботи були представлені на XLVII Науково-технічній конференції факультету комп'ютерних систем управління.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Загальні відомості

Визначення об'єктів, що рухаються є важливою задачею у сфері комп'ютерного зору. Комп'ютерний зір включає в себе отримання цифрового зображення, його обробку, аналіз і розуміння зображень, використовуючи статистичні методи і моделі побудовані за допомогою фізики, геометрії, статистики і теорії статистичного навчання. Поширення потужних комп'ютерів, доступність високоякісних камер за невеликою ціною та збільшення потреб для автоматизованого аналізу відео зумовило великий інтерес в області алгоритмів супроводження руху. Є три основні ключові кроки в аналізі відео: розпізнавання об'єкта, що рухається, відстеження об'єкта кадр за кадром, і аналіз об'єктів для визначення їх поведінки. Таким чином, використання алгоритму стеження за об'єктом є доречним в таких задачах:

- 1) Розпізнавання на основі руху, наприклад, рух людини за ходою, автоматичне розпізнавання об'єктів.
- 2) Автоматизоване спостереження, що виявляє підозрілу активність.
- 3) Взаємодія людина-комп'ютер, наприклад, розпізнавання жестів, слідування за поглядом для вводу даних тощо.
- 4) Навігація машин, що пов'язана з планування маршруту та уникання перешкод.

В найпростішій формі відстеження може бути визначений як проблема оцінки траєкторії об'єкта в площині зображення, що рухається навколо сцени. Іншими словами, відстежувач помічає постійним ярликом об'єкт, що рухається. Залежно від області, в якій використовується, відстежувач може надавати додаткову інформацію щодо об'єкта: орієнтацію, площу, розмір, форму. Відстежування об'єктів є складною задачею через такі фактори:

- 1) Нестача інформації через проекцію тривимірного світу на зображення.
- 2) Шум в зображеннях.

- 3) Складні рухи об'єктів.
- 4) Часткова або повна оклюзія об'єктів.
- 5) Зміна освітлення.
- 6) Потреба у визначенні об'єктів у реальному часі.

1.2 Особливості і проблематика

Основні проблеми, які виникають при створенні та роботі відстежувача, пов'язані з виглядом об'єкта та схожими іншими об'єктами на сцені. Вигляд інших об'єктів може накладатися на задній фон. В таких випадках, важко знайти на зображенні об'єкт, що як ми очікуємо, буде рухатися. На рисунку 1.1 зображено невизначеність кольору, що приводить до неправильної роботи відстежувача. [1].



Рисунок 1.1 – Помилкове визначення об'єктів

Також важко визначити об'єкт, вигляд якого змінився в площині об'єктива через такі фактори:

1) Зміна позиції, об'єкт що рухається змінює свій вигляд на площині зображення, наприклад якщо крутиться.

2) Зміна освітлення, напрямок, інтенсивність, колір впливає на вигляд об'єкта. Так, зміна світла в глобальному плані викликає проблеми у сусідніх сценах. наприклад, коли хмари закривають сонце змінюється навколишнє світло, а також кут

між нормаллями до поверхні об'єкта і напрямком світла. Це впливає на те, як ми бачимо зображення через лінзу камери.

3) Шум. Процес отримання зображення пов'язаний з певною долею шуму, який залежить від якості матриці камери.

4) Перекриття. Спостерігати за ціллю, коли вона частково або повністю перекрита іншим об'єктом на зображенні. Перекриття виникають коли: ціль рухається за нерухомими об'єктами, наприклад, колона чи стіл або ж інші об'єкти, що рухаються, затуляють вид на ціль.

1.3 Способи зображення об'єктів

В задачі спостереження об'єкт може бути визначений як будь-що важливе для подальшого аналізу. Наприклад, люди, машини на дорозі, риби всередині акваріуму. Об'єкти представлені їх формами та виглядом. Способи представлення, що широко використовуються для відстеження, зображено на рисунку 1.2 [3]:

1) Точки. Об'єкти представляють собою точки, або набір точок (Рисунок 1.1(b)). Підходить для стеження за невеликими регіонами на зображенні.

2) Примітивні геометричні форми. Форма об'єкта представлена прямокутником, еліпсом тощо (Рисунок 1.1(c,d)). Рух об'єкта моделюється як переміщення, афінні чи проєктивні перетворення. Підходить для стеження за твердими об'єктами.

3) Силует об'єкта та контур. Контур представляє собою границі об'єкта (Рисунок 1.1(g, h)). Силует знаходиться всередині контуру (Рисунок 1.1(i)). Підходить для стеження за нетвердими об'єктами.

4) З'єднані між собою форми. З'єднані об'єкти складаються з частин тіла, що тримаються разом за рахунок суглобів. Наприклад, тіло людини - це з'єднаний об'єкт, що складається з торсу, ніг, рук, голови та суглобів (Рисунок 1.1(e)). Відношення між частинами регулюється моделями кінематичного руху.

5) Скелетна модель. Ця модель часто використовується як форма для розпізнавання об'єктів.

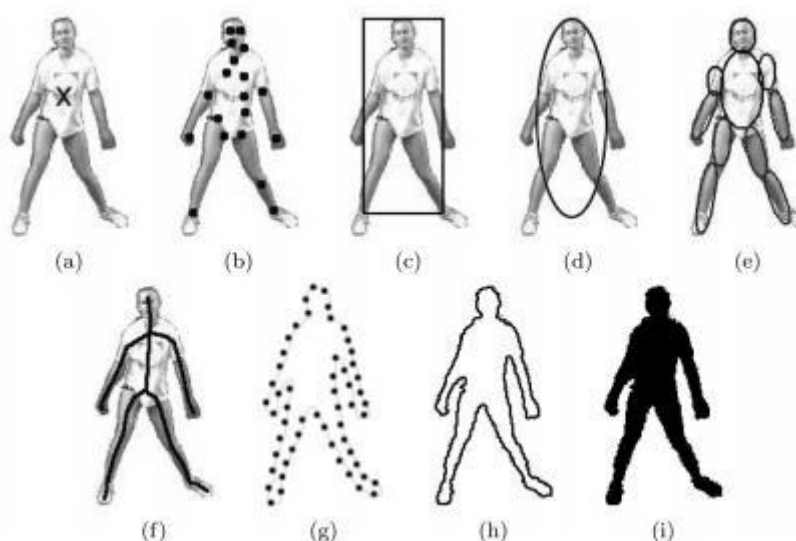


Рисунок 1.2 – Способи представлення об'єкта: а – центр об'єкта, особливі точки, с – форма об'єкта у вигляді прямокутник, d – об'єкт у формі еліпса, е – поєднання форм, f – скелетна модель, g та h – контур об'єкта, і – силует.

Також є ще декілька способів представити вигляд об'єкта. Найбільш широко використовуються такі:

1) Щільність ймовірностей вигляду об'єкта. Може бути параметризована, наприклад, за Гаусом або змішування за Гаусом, чи непараметризована, як вікно Парзена.

2) Шаблони. Формуються з примітивних форм та силуетів. Їх перевага в тому, що вони несуть як просторову інформацію, так і про зовнішній вигляд. Використовуються, якщо положення об'єкта не змінюється.

3) Активна модель вигляду. Генерується одночасно форма і вигляд об'єкта. В основному, об'єкт представляє собою набір орієнтирів. Для кожного з орієнтирів зберігається вектор вигляду: колір, текстура, градієнт.

4) Багаторакурсна модель. Ця модель описує об'єкт з різних ракурсів.

1.4 Аналіз систем відеоспостереження

Системи відеоспостереження (CCTV) – це програмно-апаратний комплекс (відеокамери, об'єктиви, монітори, реєстратори та ін. устаткування), призначений для організації відеоконтроля як на локальних, так і на територіально-розподілених об'єктах. Відеоспостереження є сьогодні невід'ємним елементом будь-якої сучасної системи безпеки [1].

На даний момент системи відеоспостереження дозволяють встановлювати відеоконтроль за об'єктами будь-якої складності. Як пристрої запису використовують як цифрові відеореєстратори (DVR), так і звичайні комп'ютери з платою відеозахоплення. Серед переваг відеореєстраторів можна виділити функції квадратора (мультиплексора), відеодетектора руху, відеомагнітофона і так далі. Також сучасні відеореєстратори забезпечені мережевим або модемним інтерфейсом, що дозволяє передавати відеодані по IP протоколу. На відміну від DVR системи побудовані на базі персональних комп'ютерів має ширший діапазон функціональних можливостей, високу продуктивність і місткість архівів, а так само звичний для користувачів ПК інтерфейс [1].

Відстеження декількох об'єктів складається з виявлення та ідентифікації об'єктів у відео. У деяких додатках, таких як робототехніка та спостереження, бажано, щоб відстеження проводилося в режимі реального часу. Це створює проблему в тому, що він вимагає, щоб алгоритм працював так само швидко, як частота кадрів відео. Сьогодні найбільш ефективні методи відстеження працюють лише з декількома кадрами в секунду, тому їх не можна використовувати в режимі реального часу. Далі, при визначенні швидкості трекера, як правило, не включати час, необхідний для виявлення об'єктів. Ми стверджуємо, що такий спосіб вимірювання швидкості не є актуальним для робототехніки або вбудованих систем, де виявлення об'єктів здійснюється на тій же машині, що і спостереження.

Один із способів запуску методу в режимі реального часу - це не дивитись на кожен кадр, а пропускати кадри, щоб відео отримало таку ж частоту кадрів, як метод

відстеження. Однак ми віримо, що це призведе до зниження продуктивності.

У цьому проекті я реалізую декілька об'єктних трекерів, слідуючи парадигмі відстеження за виявленням, як розширення існуючого методу. Він працює за допомогою моделювання руху об'єктів шляхом вирішення проблеми фільтрації та асоціювання виявлення з передбачуваними новими місцями в нових кадрах за допомогою угорського алгоритму. Використовуються три різні подібності, які використовують розташування та форму обмежувальних боксів. Порівняно з іншими трекерами в таблиці MOTChallenge, наш метод, який називається C++ SORT, є найшвидшим неанонімним поданням, а також досягає гідних балів за іншими показниками. Запускаючи нашу модель на наборі даних вибірки з різною швидкістю кадрів, ми показуємо, що продуктивність значно знижується при запуску моделі включаючи виявлення об'єктів у режимі реального часу. У більшості показників оцінка знижується на 50%, але в деяких випадках на 90%. Ми стверджуємо, що це вказує на те, що інші, повільніші методи не можуть бути використані для відстеження в режимі реального часу, але для цього потрібно конкретніше дослідження.

Сфери зображень та відео аналізу за останні роки привернули багато уваги, частково завдяки успіху глибокого вивчення моделей. Більшість найсучасніших методів використання згорткових нейронних мережі (CNNs) - глибокі мережі, які показали, що виробляють хороші результати без необхідності ручного вилучення функцій. Перехід уваги на CNNs для аналізу зображень стався в 2012 році, коли Крижевський та ін. [1] переконливо виграв ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) [2], змагання, використовуваного для моделей аналізу зображень із порівняльним оцінкою. Завдання полягало в класифікації зображень, де метою було позначити кожне зображення одним із декількох класів. Наприклад: сповістити нас, якщо на малюнку є кішка або собака. З того часу були створені більш точні моделі. Комп'ютери 2015 року могли б перевершити людей у цьому завданні [3]. Використання CNNs не обмежується розпізнаванням зображень, але успішно прийнято в таких завданнях, як виявлення об'єктів (локалізація об'єктів заданих класів), семантична сегментація (маркування кожного пікселя зображенні) та

підписи зображень (створюючи описовий текст вмісту) [4, 5].

Відеоаналіз тісно пов'язаний з областю аналізу зображень, у відео є кілька зображень, складених у часі. Були вирішені подібні виклики, як і в аналізі зображень, включаючи класифікацію відео та виявлення об'єктів у відео, а також завдання, що ексклюзивно для відео, наприклад, відстеження об'єктів (ідентифікація об'єктів у кількох кадрах), прогнозування траєкторії (оцінка шляхів об'єктів) та розпізнавання дій (класифікація дій у послідовності відео). Як і в аналізі зображень, CNNs дають чудові результати для цих завдань [6, 7].

Наявність точних моделей зображень та відео аналізу є основою для автономних транспортних засобів та роботів. Моделі виявлення об'єктів можуть давати можливість самостійно керувати автомобілями, щоб уникнути зіткнень, і деякий рівень розуміння ситуації потрібен роботам, щоб реагувати на оточення. При збільшенні використання безпілотних літальних апаратів (БПЛА) - загальновідомі як безпілотники – для таких завдань, як спостереження, доставка та пошуково-рятувальні роботи [8]. Важливо розробити інструменти, придатні для цих застосувань та адаптовані до БПЛА. Часто так і є необхідні для ідентифікації та відстеження людей та інших рухомих об'єктів. Наприклад, у завданнях спостереження може бути суттєвим відстеження людей, щоб виявити незвичну поведінку. На жаль, не всі моделі відеоаналізу використовують БПЛА. Найпотужніші моделі вимагають обчислювального обладнання високого класу, але для БПЛА, які повинні бути легкими та недорогими, обладнання обов'язково обмежується.

1.5 Відеокамери, системи відеоспостереження

На сьогоднішній день система відеоспостереження – це, насамперед, сучасні камери відеоспостереження, які бувають вбудованого, зовнішнього і бездротового типу.

Весь світ вдало використовує відеоспостереження, Україна тут не стала

виключенням. Зараз у нашій країні можна купити відеоспостереження для різних цілей, наприклад, відеоспостереження будинку, супермаркетів, організацій, зовнішнє відеоспостереження територій та ін. Для кожного об'єкта підбирається найбільш підходяща програма для відеоспостереження[1].

Система відеоспостереження, а особливо приховане відеоспостереження, допомагають підтримувати порядок, дисципліну і вирішувати конфлікти, непорозуміння, кримінальні ситуації.

Також, з кожним днем все більшою популярністю користується безпроводне відеоспостереження, оскільки його швидко і легко встановити на будь-якій поверхні і можна переміщати з одно місця в інше.

Часто системи відеоспостереження входять до складу охоронного комплексу, разом з охоронною сигналізацією, пожежною сигналізацією і т.д. Відеоспостереження буває зовнішнє (усередині кімнат, кабінетів, споруд) і внутрішнє (стеження за підходами до будівлі, двором, воротами).

Відеокамери є "очима" будь-якої системи відеоспостереження, і від правильного їх вибору залежатиме, зображення на екрані монітора. Слід чітко зрозуміти, що повинна бачити відеокамера, в якому місці вона буде встановлена, способі живлення і в яких умовах працюватиме. Виходячи з цього і слід вибирати камери для відеоспостереження. Для зручності установки, камери бувають різного зовнішнього вигляду.

Стандартні камери – найпоширеніший тип камер. Встановлюються за допомогою кронштейна на будь-яку поверхню (стіни, стелі і так далі) Для установки у вуличних умовах, поміщаються в захисний термокожух. В таких відеокамер є одна перевага перед іншими камерами – до стандартних відеокамер можна “прикрутити” практично будь-який об'єктив з потрібним вам кутом огляду і наближенням.

Купольні (стельові) камери. Встановлюються виключно на стелю, при правильному виборі вони досить естетично виглядають і не кидаються в очі. Бувають чорного і білого кольору.

Мініатюрні камери. Вони бувають різних форм (циліндрова, овальна,

прямокутна і так далі). Часто застосовуються там, де потрібно зберегти дизайн інтер'єру або не залучати уваги.

Модульні (безкорпусні) камери. Є відкритою платою з мікросхемами і об'єктивом, яку можна вбудувати в потрібний корпус. Найчастіше використовуються для прихованої установки, або установки в малогабаритні термокожухи [2].

Закамуфльовані камери. Найчастіше виглядають як охоронний або димовий датчик. У датчик встановлені модульні відеокамери. Використовуються для прихованого відеоспостереження. Вуличні камери. Призначені для установки у вуличних умовах. Такі камери мають герметичний корпус з підігріванням, сонцезахисний козирок і найчастіше вбудоване інфрачервоне підсвічування.

Поворотні камери. Дозволяють віддалено сидячи за пультом на посту охорони, наближати/віддаляти об'єкт спостереження і повертати в різні сторони. Мають режим патрулювання. Незалежно від зовнішнього вигляду всі камери відеоспостереження володіють одним і тим же набором основних технічних характеристик [2].

1.6 Технічні характеристики відеокамер

Розподільна здатність вимірюється в телевізійних лініях (ТВЛ). Стандартним вважається дозвіл 420 ТВЛ, високий дозвіл – 600 ТВЛ. Для більшості випадків досить відеокамери із стандартним дозволом. Високий дозвіл потрібний тоді, коли потрібно розрізняти дрібні деталі, наприклад, цифри на банкнотах, номери машин і так далі. Також слід зазначити, що високий дозвіл 600 ТВЛ досяжно лише в чорно-білих камерах. В кольорових камер високим дозволом вважається 480 ТВЛ [3]. Чутливість вимірюється в люксах (Лк) і характеризує здатність камери “бачити” при слабкому освітленні. Чим менше цей параметр в камери, тим краще. Чорно-білі камери мають велику чутливість, тобто краще бачать в темряві. Стандартна чутливість в чорно-білих камер – 0,05 Лк, в кольорових – 1 Лк.

Функція день/ніч, за наявності даної функції у відеокамери при малій

освітленості зображення автоматично конвертується з кольорового в чорно-біле. Режим компенсації заднього засвічення дозволяє камері вибрати кращі умови зйомки в будь-якій обстановці і автоматично встановлює необхідний рівень контрастності на яскравих ділянках. Таким чином, користувач завжди отримує чисту картинку з найдрібнішими деталями і прекрасною контрастністю. У цьому режимі рівень яскраво освітленого фону компенсується електронним способом, завдяки чому замість силуета, що зазвичай виходить в таких умовах, забезпечується добре освітлене зображення.

Більшості людей подобається кольорове відео і це природно, на деяких об'єктах краще використовувати чорно-білі відеокамери. В той же час бувають випадки, коли переважне кольорове зображення. Наприклад, спостереження за автомобілями або потоком людей. У будь-якому випадку це справа вибору, який інколи обмежується особливостями об'єкту і ресурсами.

Модульна (безкорпусна) камера складається з відкритої плати з мікросхемами і об'єктиву. Безкорпусні камери на перший погляд виглядають ненадійними і здається, що варто таку камеру узяти в руки відразу щось зламається. Але це лише здається, насправді вони досить надійні і їх не так просто вивести з робочого стану. Їх використовують, у панелях виклику відеодомофонів, наприклад в панелях Comtex [4].

Безкорпусні камери з точковим об'єктивом "pin-hole" використовують для прихованої установки, наприклад в корпусі ГИК – датчика і тому подібне. Можна звичайно використовувати в цьому випадку і корпусні камери, але навіщо, якщо передбачається, що її все одно не буде видно. В більшості На базі безкорпусних камер виробляють і безкорпусні дверні вічка, які імітують зовні звичайне вічко і що розширює кут огляду аж до 170 градусів. Відеокамера підключається до телевізора безпосередньо через вхід тюльпан або SCART5. При цьому камеру треба живити від блоку живлення. Перемикання між режимами телевізор/відеокамера здійснюється аналогічно перемиканню телевізор/відеомагнітофон (TV / AV). Максимальна допустима відстань між

телевізором і відеокамерою залежить від типу використовуваного дроту. Практика показала, що при використанні мультикабеля, дріт можна упевнено тягнути на 100-150 метрів. При великих відстанях необхідно використовувати відеопідсилювач [5].

1.7 Спеціальна відеоапаратура

Відеореєстратор (англ. Digital Video Recorder, DVR, цифровий відеореєстратор) — пристрій, призначений для запису, зберігання і відтворення відеосигналів, а в деяких випадках і аудіосигналів що поступають від аналогових відеокамер і мікрофонів в системах відеоспостереження і аудіоконтролю. Залежно від поставлених завдань відеореєстратор може використовуватися для вирішення різних завдань, у тому числі: для відеоспостереження за відвідувачами в приватних будинках, офісах, магазинах; для відеоконтроля касових операцій на робочих місцях касирів в магазинах (для аналізу відповідності відеоданих, даних касових терміналів і чеків); для організації системи обліку і контролю автотранспорту на парковках і КПП підприємств; для автоматичної реєстрації і контролю переміщень залізничних вагонів по території заводів, комбінатів, паливно-заправних комплексів, сховищ; при організації територіально-розподілених систем відеоспостереження, у тому числі з єдиним центром моніторингу [6].

Відеореєстратор є складним електронним пристроєм, схожим по будові з комп'ютером або відеосервером і містить в своєму складі АЦП, процесором, жорстким диском і інші компоненти. Для управління відеореєстратором на ньому встановлена спеціалізована операційна система. Перед записом оцифровані відеозображення як правило піддаються компресії з метою зменшення займаного ними об'єму на жорсткому диску. Практично всі відеореєстратори можуть працювати як з монохромними, так і з кольоровими відеозображеннями. Багато відеореєстраторів мають можливість підключення до комп'ютерної мережі для передачі відеозображень на комп'ютери віддалених користувачів.

Відеореєстратори характеризуються такими параметрами, як: функціональність (симплекс, триплекс, пенталекс); число вхідних відеоканалів; сумарна швидкість запису; дозвіл; використовуваний тип компресії; кількістю записуваної інформації до перезапису; можливості підключення зовнішніх накопичувачів інформації; можливістю підключення аудіоканалів; можливістю підключення ір-відеокамер; можливістю роботи в мережі і багатьма іншими параметрами. У системах відеоспостереження відеореєстратори, практично повсюдно, замінили собою відеомагнітофони, що раніше застосовувалися для цих цілей, і квадратори, основним конкурентом відеореєстраторів в системах відеоспостереження зараз є відеосервери . Зараз окремі різновиди DVR виділилися в підгрупи, які мають власні назви: NVR — мережевий відеореєстратор (робота лише з ір-відеокамерами), HDVR — гібридний відеореєстратор (робота з аналоговими і ір-відеокамерами), PC-based DVR (відеореєстратор на базі ПК)stand alone DVR (робота тільки з аналоговими відеокамерами). Car DVR — відеореєстратор, призначений для установки в автомобілях, вантажівках, катерах і інших транспортних засобах

Мережеві відеореєстратори або NVR (англ. Network Video Recorder, мережевий відеореєстратор) призначені для роботи в ір-системах відеоспостереження. На відміну від звичайних DRV, NVR отримують відеодані вже в стислому вигляді по мережі Ethernet. Дані можуть поступати з ір-відеокамер або з аналогових відеокамер, що підключаються через спеціальні адаптери (типу: «композитний сигнал — Ethernet»). Особливістю NVR є те, що вони можуть працювати лише з обмеженим списком моделей ір-відеокамер, оскільки в даний час стандартизація їх інтерфейсів мережевого обміну ще не поширена. Переваги віддалений доступ до відеоданих по локальній мережі або Інтернет (через спеціальний софт або web-інтерфейс) легко нарощувана архітектура підключення до локальної мережі в довільному місці. Недолік високе навантаження на локальну мережу.

Типові характеристики мережевого DVR: кількість відеоканалів (ІР відеокамер): 4/8/16 кількість HDD: 2/4 (підтримка RAID) підтримувані формати відеостискування: MJPEG, Mpeg4, H.264 наявність аудіо/відео інтерфейсів для

виведення інформації з відеокамер на локально підключений телевізор.

Як правило, в системах відеоспостереження використовується чимала кількість відеокамер, встановлених для контролю різних об'єктів. Якщо зображення від всіх відеокамер виведене на монітор і перемикається уручну, то час між переглядом зображення з однієї і тієї ж камери сильно збільшується, що приводить до зменшення ефективності роботи всієї системи. Квадратори призначені для одночасного виводу на екран монітора зображення від 4-х відеокамер. Бувають також двохсторінкові квадратори, що дозволяють виводити зображення від 8 камер двома сторінками, – по 4 камери в кожній. Зазвичай екран монітора при цьому ділиться на 4 частини (режим "квадро"). Проте, як правило, взаємне розташування зображень від різних камер можна міняти – їх можна переглядати в режимах "картинка в картинці", "картинка над картинкою" або по черзі на повному екрані. Часто квадратори мають вбудований датчик руху, звукову сигналізацію в разі втрати сигналу від камери. Відеокомутатор - це простий пристрій, що дозволяє передавати зображення від декількох (зазвичай до 8) відеокамер на відеомонітор по черзі з регульованою затримкою (від 0,5 до 60 сек). Можуть працювати в режимі ручного перемикання. Деякі моделі забезпечуються тривожними датчиками, при спрацьовуванні яких на екран автоматично передається зображення з відповідної зони спостереження.

Мультиплексування – найбільш ефективний спосіб безперервного запису зображення від декількох джерел на один носій. Мультиплексний запис – це запис повних кадрів послідовно (кадр за кадром) від кожної з підключених камер. Мультиплексний запис забезпечується вживанням процесора цифрової обробки зображення з буферною кадровою пам'яттю, в яку записується зображення від відеокамер. Процесор з пам'яттю служить в цьому випадку коректором тимчасової. До появи мультиплексорів для цієї мети служили фреймвітчери – перемикачі відеосигналу, в яких перемикання здійснюється у момент кадрового синхроімпульсу. При цьому потрібно, щоб всі відеокамери були синхронні і синфазні одна з одною. Процес формування вихідного сигналу для запису називається кодуванням. При цьому в невидимі рядки кадрового бланка записується службова інформація (ID

камери, дата, час, мітки тривоги і тому подібне). Зворотний процес при відтворенні називається декодуванням. Основним недоліком послідовного мультиплексного запису є збільшення періоду оновлення зображення від кожної із записуваних камер (ефект мультиплікації). При установці магнітофона в режим тривалого запису цей ефект може привести до втрати кошовної інформації. Для часткового усунення цього недоліку практично у всіх моделях мультиплексорів застосовується спосіб динамічного розподілу часу запису, в основі якого лежить аналіз змін в зображенні. Алгоритми виявлення змін можуть бути різними, від простої яскравості до складної, з аналізом зображення у фазовому просторі. При виявленні змін в зображенні від камери, частота запису зображення від цієї камери збільшується, що зменшує вірогідність пропуску важливих подій. У останніх моделях мультиплексорів виходи детекторів активності в кожному каналі обробки виводяться на зовнішній роз'єм і можуть бути використані для інших вживань [7].

Оскільки дуже часто камери відеоспостереження доводиться встановлювати в умовах екстремальних для таких пристроїв, наприклад, при низьких температурах, підвищеній запиленій, високій вологості, необхідно передбачати захист від цих чинників. Таким захистом і є термокожух. Камера кріпиться усередині термокожуха, де їй забезпечується герметичність і необхідний температурний режим, а також підводиться живлення і телевізійний кабель. При установці систем відеоспостереження часто виникає проблема здобуття хорошого зображення території, що переглядається, в умовах недостатньої освітленості на об'єкті. Це повною мірою можна віднести як до закритих приміщень з вимкненим або черговим освітленням, так і до об'єктів на відкритому повітрі. Вирішитись ця проблема може різними способами: установкою дорогих високочутливих камер, організацією штучного освітлення або установкою систем інфрачервоного підсвічування.

Практично всі сучасні відеокамери виконані на основі світлочутливих ПЗС матриць. Більшість "недорогих" камер мають чутливість не більше 0.1 люкс., що дає прийнятну якість зображення об'єкту на моніторі в ясну ніч при повному місяці. У багатьох випадках оптимальне використання звичайного штучного освітлення

видимого діапазону. Це переважно хоч би тому, що дозволяє відеокамері працювати в максимумі її чутливості. Але, із зрозумілих причин, це неприйнятно при веденні прихованого відеоспостереження на об'єкті. У таких випадках найбільш доцільною є установка системи інфрачервоного підсвічування. В порівнянні з людським оком чутливість чорно-білих камер істотно зрушена в інфрачервону область, що дозволяє використовувати в умовах недостатньої освітленості спеціальні інфрачервоні пристрої підсвічування. Інфрачервоне випромінювання не видно людському оку, проте прекрасно фіксується самою відеокамерою.

При організації системи інфрачервоного підсвічування необхідно враховувати, що самі пристрої підсвічування діляться на двох основних типів: з видимими і не видимими інфрачервоними випромінювачами [7].

Таким чином, у даному розділі було проаналізовано системи відеоспостереження і розглянуто основні технічні характеристики відеокамер.

1.8 Визначення проблеми

Коли відео містить кілька об'єктів, що рухаються, які ми хочемо відстежувати, ми називаємо це відстеженням декількох об'єктів. Див. Рисунок 1.1 для ілюстрації. У певному сенсі завдання - це розширення виявлення об'єктів, оскільки, крім виявлення об'єктів, нам потрібно з'єднати виявлення між кадрами, щоб отримати послідовне відстеження. Виявлення об'єктів все ще залишається невирішеною проблемою, а найпотужніші методи обмежені швидкістю [9]. Додавання можливостей відстеження поверх детектора зазвичай уповільнює алгоритм. Через це багаторазове відстеження об'єктів важко здійснити в режимі реального часу, оскільки найкращі алгоритми можуть аналізувати лише кілька кадрів в секунду в кращому випадку навіть на потужному обладнанні [10]. Щоб такі алгоритми запускалися в режимі реального часу, потрібно було б пропустити кілька кадрів, щоб запобігти постійно зростаючій затримці. Наскільки нам відомо, не було проведено великої роботи, щоб дослідити,

як вибірка кадрів вплине на ефективність відстеження. В MOTChallenge, щорічна конкуренція та орієнтир багаторазового відстеження об'єктів, мало стимулів робити моделі, здатні працювати в режимі реального часу.

У цьому проекті ми хочемо дослідити, як функціонуватиме відстеження кількох об'єктів у реальному часі. Для цього реалізована система відстеження декількох об'єктів, яка оцінюється на існуючих наборах даних для забезпечення порівняння з іншими роботами. Оскільки мета полягає в тому, щоб метод запускався в режимі реального часу, важливим фактором є швидкість трекера. Через це вирішено, що простий підхід віддається перевазі над повільним, але потенційно більш потужним один. Порівнюючи роботу з громадськими лідерами, ми отримуємо об'єктивну оцінку нашого методу. Щоб зрозуміти, наскільки потужніші моделі будуть працювати у режимі реального часу за допомогою кадрів вибірки, наша модель працює з різними конфігураціями та різною швидкістю вибірки на великому наборі даних. Ефект від продуктивності від зниження частоти кадрів вимірюється з надією отримати уявлення про те, як інші моделі зможуть дійти за таких обставин.

Як випадок використання для відстеження в реальному часі, можна уявити, що повна модель використовується на невеликих БПЛА з обмеженим обладнанням. БПЛА передають інформацію бездротово до наземної станції управління, а через низьку пропускну здатність та запобігання латентності відеокадри не передаються. Натомість усі обчислення проводяться на БПЛА, а вилучена інформація надсилається у вигляді повідомлень через рівні проміжки часу. Потрібно, щоб система працювала в режимі реального часу та в режимі он-лайн, тобто для того, щоб виходити стабільно та без особливих затримок. Це особливо складно з огляду на обмежене обладнання БПЛА, яке перешкоджає використанню пам'яті та обчисленню важких моделей.

У програмі MOTChallenge користувачі надають текстові файли виявлення для кожного кадру в наборі даних, як спосіб переконатися, що всі знаходяться на рівних місцях, а можливість відстеження насправді вимірюється на відміну від якості виявлення. Це мотивує використання парадигми під назвою відстеження за виявленням (див. Розділ 2.2), яка є агностичною для типу використовуваного

детектора. Наш метод також перевірений на Okutama-Action [11], який має перевагу у наданні двох попередньо підготовлених моделей: одна для виявлення об'єктів та одна для розпізнавання дій, а також виявлення, які вони видають у вигляді текстових файлів. Злегка змінивши вихід моделі розпізнавання дій, її можна використовувати для звичайного виявлення об'єктів. Перевага від наявності двох різних моделей полягає в тому, що ефективність відстеження може бути виміряна, коли забезпечуються виявлення різної якості.

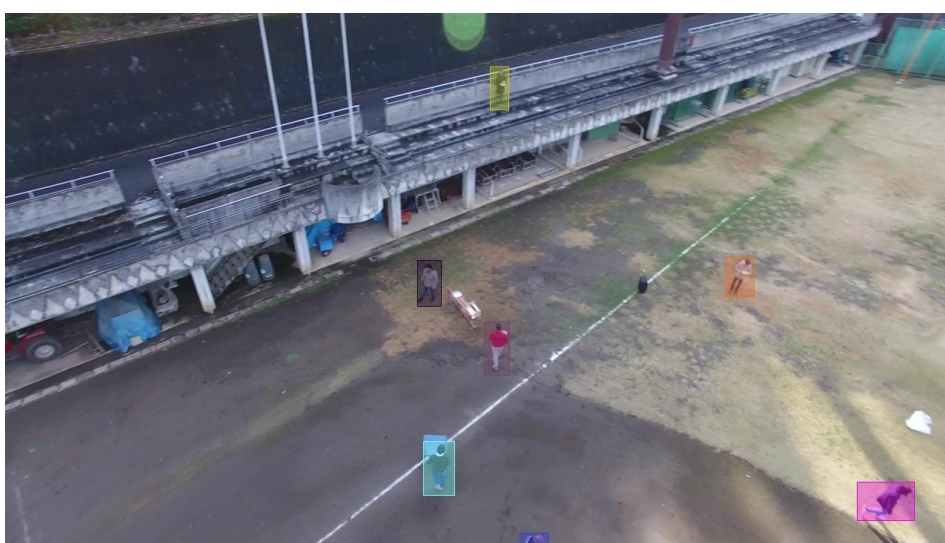


Рисунок 1.1 — Відстеження семи осіб у відео

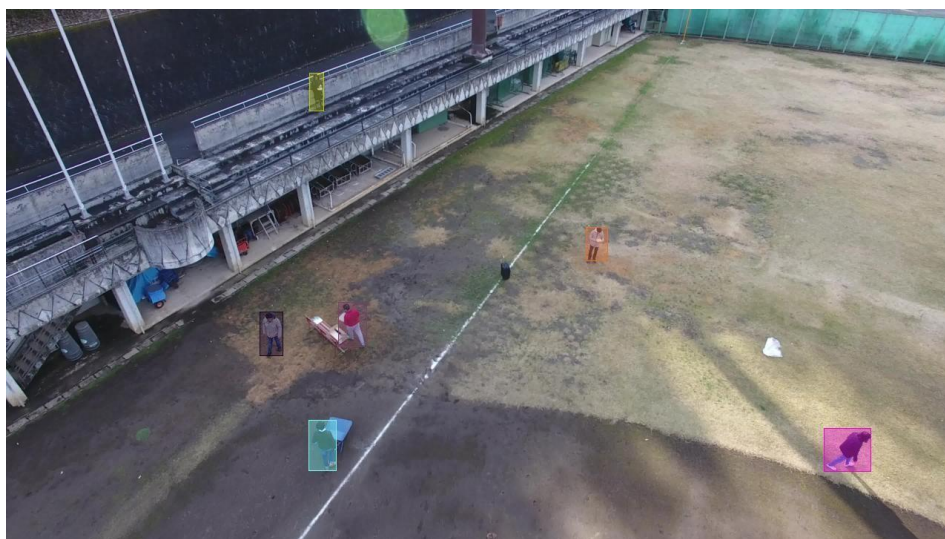


Рисунок 1.2 — Відстеження семи осіб у відео (одна з яких залишає сцену між

відображеними кадрами). Кожна людина укладена у кольоровий обмежувальний бокс , у якому кольори представляють особистість. Потенційні збої включають змішування ідентичності (кольорів) або відсутність деяких людей повністю.

1.9 Дослідницьке питання

У цій тезі ми дослідимо, чи можуть існуючі онлайн моделі відстеження за виявленням успішно використовуватися для відстеження кількох об'єктів у режимі реального часу. Щоб відповісти на це, ми запровадимо систему відстеження та оцінимо її ефективність у різних конфігураціях. Для подальшого дослідження відстеження в реальному часі ми також виміряємо, як частота кадрів відео впливає на можливість відстеження, оскільки пропуск кадрів - це спосіб змусити більшість моделей працювати в режимі реального часу. Сподіваємось, що отримані результати можуть спрямовувати майбутні дослідження щодо відстеження в реальному часі або принаймні підвищувати інтерес до цієї проблеми. Теза буде розглянута успішно або якщо ми знайдемо модель, яку можна використовувати для відстеження кількох об'єктів у реальному часі, або якщо ми отримаємо уявлення про те, що потрібно для досягнення цього.

1.10 Обмеження

Оскільки мета - відстеження в режимі реального часу, розглядаються лише методи стеження з низькою швидкістю виводу. Це означає, що методи, що використовують функції зовнішнього вигляду для асоціації об'єктів між кадрами, не враховуються, оскільки обчислення таких особливостей відбувається досить повільно. Натомість використовуються лише геометричні та позиційні властивості. Крім того, для досягнення модульності, необхідної для використання або моделей виявлення об'єктів, або розпізнавання дій, не розглядається модель, яка вимагає конкретного типу детектора об'єктів. З тієї ж причини методи, які включають передбачуване місце розташування від відстеження, щоб направляти або вдосконалювати детектор об'єкта не використовуються. Метою магістерської роботи є не запропонувати нову модель відстеження, а використовувати та поєднувати ідеї з

існуючих методів та намагатись їх ефективно реалізувати.

2 ВІДСТЕЖЕННЯ ДЕКІЛЬКОХ ОБ'ЄКТІВ

2.1 Відстеження багатьох об'єктів

Так як запропонована назва «відстеження багатьох об'єктів» складається з відслідковування об'єктів, які рухаються (рухомі об'єкти). Формальний опис буде таким: для кожного кадру на відео локалізувати та ідентифікувати усі об'єкти, які нас цікавлять, так щоб тотожності були послідовними протягом усього відео. Іншими словами, хороша модель повинна правильно виявити об'єкти у кожному кадрі і забезпечити послідовне маркування. Це представлено на рисунку 1.1. Ризики зростають, коли об'єкти є частково або повністю закритими (невидимими на відео) або тимчасово покидають поле зору на камері. Так як в ідеалі об'єкти повинні зберігати їх попередній ідентифікатор при кожній наступній появі. Більш того, об'єкти, чії траєкторії перетинаються, можуть збити з пантелику модель і спричинити помилкове надання ідентифікаторів.

Існують різні сценарії, які потребують різних типів моделей. Важливою відмінністю є порівняння онлайн з офлайн моделями [12]. В онлайн моделі на відео ми отримуємо вхідні дані на основі кадр за кадром (frame-by-frame basic), і видає вихідні дані для кожного кадру. Це означає, що в додаток до поточного кадру може використовуватися інформація лише з попередніх кадрів. З іншого боку офлайн моделі мають доступ до всього відео і це означає, що може використовуватися інформація як з попереднього так і з наступного кадру. Завдання може бути представлене проблемою оптимізації, де ціллю є знайти набір траєкторій (шляхів) щоб мінімізувати деякі вилічесні втрати функцій. Цю проблему може вирішити лінійне програмування [13,14] і методи оптимізації (k-shortest) найкоротших траєкторій (шляхів) [15]. Так як офлайн відстежування має доступ до більшої кількості інформації, то ми можемо очікувати кращого виконання задач від цих моделей. Однак, зауважимо, що використання в реальному часі вимагає безпосередньо онлайн

методів, так як наступні кадри є недоступними. Для обговорення моделей в реальному часі дивіться пункт 2.3. Якщо інше не зазначено, то решта цього розділу присвячена онлайн моделям.

Моделі можуть бути поділені на одиночні і множинні класи моделей [16]. В минулому усі відстежувані об'єкти є одного і того ж класу. Наприклад : людина, в той час , як множинні класи представлені пішоходом велосипедом і авто. З одного боку множинні класи ознайомлюють з проблемою класифікації кожного об'єкту. З іншого боку це може відбуватися як розрізнення певних рис, коли траєкторії об'єктів окремих класів пересікаються.

Обширне опитування з 2006 року систем об'єктів відстежування [17] показало що до цього часу дослідження були зосереджені на знаходженні хороших особливостей представлення об'єктів і руху моделей щоб покращити якість відстеження. Було зазначено, що більшість методів допускають деякі обмеження в налаштуваннях, які дуже полегшують відстежування. Наприклад , якщо статична камера , тоді є можливим знайти всі рухомі об'єкти за допомогою застосування субстракції заднього плану[18]. Іншими припущеннями були те, що об'єкти мали високий контраст з заднім планом , рухалися плавно та без оклюзії. Уілмаз [17] ідентифікував ці припущення досить обмеженими що стосується застосування у реальному світі, зазначаючи що поки вказані припущення підвищували продуктивність, вони робили моделі надто специфічними.

2.2 Відстежування за допомогою виявлення

Опитування Уілмаза [17] були опубліковані перед глибоким вивченням отриманої уваги в аналізі картинок у 2012 році, коли Крижевський [1] виграв змагання по широкому розпізнаванню образів в ImageNET в номінації ILSVRC використовуючи поглиблене вивчення. Сьогоднішня згортова нейронна мережа(CNN) дуже перевершує попередні методи в завданні класифікації картинок.

Огляд згорткової нейронна системи (CNNs) и бази представлений у пункті 2.2.1. Для виявлення об'єктів три бази даних зазвичай використовуються для орієнтування : ImageNet[2], Pascal VOC [19], і MS COCO [20]. Найкраще виконання моделей для усіх трьох випробовуваннях використовують CNNs відмічаючи, що теперішня можливість виявлення є кращою, аніж в 2006 році. Це в свою чергу веде до відновлення інтересу в окремій парадигмі відстежування , яка називається відстежування за допомогою виявлення.

Хоча усі системи відстежування потребують виявлення об'єктів а деякій стадії [17], методи відстежування за допомогою виявлення формують чітку відмінність між виявленням та відстежуванням об'єктів. Загальною ідеєю для кожного кадру є перш за все локалізувати усі об'єкти використовуючи детектор об'єктів, потім співставити виявлені об'єкти між кадрами , використовуючи такі ознаки як місцезорозташування та зовнішній вигляд. Авідан [24], раніше використовував цей підхід користуючись попередньо підготовлену сапорт векторний механізм (SVM), для визначення транспортних засобів та оптичних потокових рівнянь щоб сполучити виявлення між кадрами. Це відзначають Бевлей у книзі [25] та Юу [26], що ефективність моделей відстеження за виявленням сильно залежить від точності моделі виявлення, що вказує на те, що детектори об'єктів на основі CNN можуть значно підвищити ефективність відстеження навіть у простих моделях відстеження. Як обговорюється в розділі 2.3, наявність потужного детектора може призвести до зниження частоти кадрів у сценаріях реального часу.

Загальна методологія полягає в поділі відстеження на дві фази: прогнозування локального об'єкта і відповідність виявлення та прогнозування [25]. Тобто для кожного нового кадру повна модель відстеження виконує наступне: 1. вона виявляє об'єкти, що нас цікавлять; 2. передбачає нові місця розташування об'єктів з попередніх кадрів;

3. пов'язує об'єкти між кадрами за подібністю виявлених та передбачуваних місць. Хоча деякі методи відрізняються від цього підходу незначно, наприклад шляхом узгодження об'єктів у декілька етапів на основі достовірних значень [27] або

підтримання численних гіпотез активними [28, 29], більшість робіт відрізняються просто виконанням цих трьох кроків окремими шляхами.

2.2.1 Виявлення

Метою об'єкту є його виявлення, яке впливає з назви, тобто виявити на зображенні всі об'єкти певного класу. Крім того, може існувати декілька класів, і в цьому випадку також потрібно правильно класифікувати кожен об'єкт. Вхід до детектора об'єктів - це зображення, і вихід - це список обмежувальних боксів, з мітками у випадку декількох класів. Обмежувальне поле зазвичай представлено піксельними координатами у верхньому лівому та нижньому правому кутах обмежувального поля або координатами центру, а також шириною та висотою поля. Крім того, більшість детекторів об'єктів забезпечують достовірність кожного боксу із зазначенням надійності виявлення. Загальний показник продуктивності детектора об'єктів - середня точність для всіх класів (див. розділ 2.5). Як було сказано вище, методи виявлення об'єктів за допомогою CNNs є сучасними, перевершуючи попередні методи, такі як SVMs.

Конволюційна нейронна мережа CNN - це тип штучної нейронної мережі, призначений використовувати переваги структур у зображеннях та подібними даними. У 1990 році Лі Кан та ін. [30] використовував невеликий CNN для розпізнавання рукописних цифр - це було першим успішним використанням CNN. Мережа, названа LeNet, складалася з 5 шарів: 2 набори згорткового шару, за яким слідує шар об'єднання, і нарешті повністю пов'язаний шар. Показана архітектура LeNet на малюнку 2.1. Хоча сучасні CNN набагато глибші (тобто мають більше шарів), загальна структура змінних шарів згортки та об'єднання, за якими слідує повністю з'єднаний шар, залишається [4].

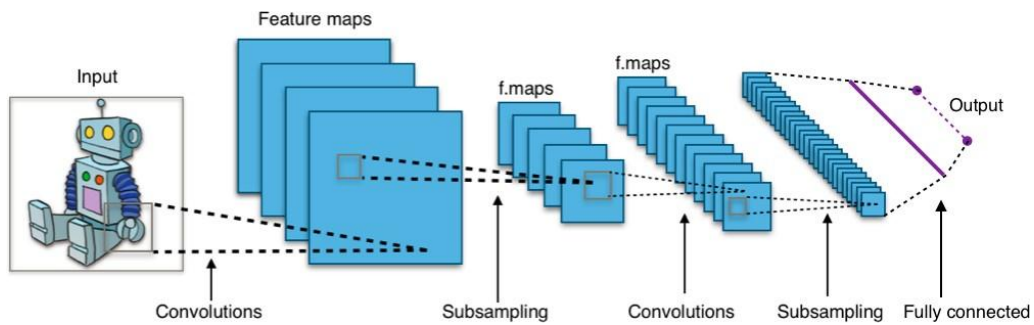


Рисунок 2.1 — Згорткова нейронна мережа з 5 шарами: 2 згорткових шари, за яким слідує шар об'єднання, і повністю пов'язуючий шар.

Повністю з'єднані шари мають вагу, пов'язану з кожною парою входу та виходу. Вихід обчислюється як внутрішній добуток ваги і входу. Ці ваги повинні бути вивчені, і оскільки кількість ваг пропорційна кількості входів та кількості виходів, кількість параметрів швидко зростає для повністю з'єднаних шарів. Зворотні шари, з іншого боку, мають невеликий набір ваг, що називаються фільтрами. Щоб обчислити вихід, кожен фільтр проходить по входу, і в кожному положенні обчислюється внутрішній продукт між фільтром і входом у цьому місці. Це має перевагу при обміні ваги, що різко зменшує кількість параметрів. Крім того, як тільки фільтр навчився захоплювати певну структуру, він може знайти його в будь-якому місці зображення, оскільки фільтр застосовуватиметься у всіх положеннях. Це бажано для зображень, оскільки одні й ті ж об'єкти та структури можуть з'являтися де завгодно. Пізніше, об'єднанні шари використовуються для зменшення розмірності даних.

Як і в згорткових шарах, об'єднання проводиться в кожній позиції вводу, хоча зазвичай без перекриття. Існують різні варіанти, але при максимальному об'єднанні, яке найчастіше використовується, результат обчислюється як максимальне значення введення. Зокрема, шари об'єднання не мають вивчених параметрів [4].

Популярними високоефективними архітектурами CNN, які використовуються для виявлення об'єктів, є R-FCN [31], швидкий R-CNN [32] та SSD [33].

3 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 C++SORT

Наша робота слідує парадигмі відстежування за допомогою виявлення, яка містить в собі завдання виявлення об'єктів у кожному кадрі окремо від відстежування. Зазвичай так як у MOTChallenge ми не пов'язані з отриманням виявленням, але припускаємо, що ми маємо спосіб вирізаням їх і використовувати виявлення як основу до алгоритму відстежування. Щоб прискорити початок та розвиток моделі відстежування за допомогою виявлення. Виявлення часто обчислюється в режимі офлайн заздалегідь і зберігається як текстові файли для швидкого завантаження в трекер. В нашій роботі трекер не відслідковує рід виявлення, навіть у випадку коли вони отримуються в режимі онлайн, і не робить спроб керувати виявленням основаних на поточних відстежуваннях.

Ми використовуємо метод Бевлі [25], Який називається SORT – Simple Online and Realtime Tracking. Однак, ми розширюємо метод поволі розглядаючи інші прогнозовані методи, такі як фільтр Калмана та інші. Щоб досягти методів та їх результатів, ми звертаємося до моделі C++SORT, яка фільтр теорій представлені у розділі 2.2 та його підрозділах.

Коротко кажучи ідеєю є відслідковувати кожен об'єкт, моделюючи його рух від кадру до кадру за допомогою передбачення, яким може бути Калмана або частотний фільтр(розділ 2.2.2). Для кожного кадру відбувається наступне: об'єкт відстежується, нові місця знаходження вже відстежених об'єктів передбаченні використовуючи алгоритм виявлені та відстежені об'єкти сполучені між собою співставлені за допомогою подібностей, які основані на обмежувальних боксах. Алгоритм передбачення оновлює пов'язані виявлення та досягнені позиції перетворюються у відстеження. В додаток, новий алгоритм видаляє непарні виявлення та невикористовувані алгоритми передбачення. Метод представлений більш детально

у алгоритмі 1.

3.1.1 Виявлення

У MOTChallenge, виявлення подаються як текстові файли, які в кожному ряду містять наступну інформацію з виявлення: номер кадру, позиція повісі x , позиція по повісі y , ширина, висота, і код (1 або 0). Подані виявлення не є ідеальними, мається на увазі, що позиції можуть злегка відхилятися від осі і що деякі інші незначні похибки можуть також мати місце. Конфіденційне значення використовується для того, щоб визначити виявлення, які використовуються трекером. За допомогою мережевого пошуку було знайдено значення порогу 0.4, яке задовольняє алгоритм 1. В додаток до вище зазначеної інформації подається правдивий ID, який очевидно є єдиним доступним для навчального набору. В Okutama-Action, тип CNN який називається SSD [33] навчає і використовує виявлення об'єктів. В завданні виявлення людини, він має лише вихід, але розпізнавання дій має кожен діючий тип. Однак, так як людина є єдиною дійовою особою і кожна людина маркована дією в кожному кадрі, вихідні мережеві дані для розпізнавання дії можуть використовуватися для відслідковування, якщо ігнорується показник дії. Мережа, яка навчає розпізнавання дії є повільнішою та менш точною, ніж інша, так як розпізнавання дії є важчим завданням ніж виявлення людини.

Приклад алгоритму :

Algorithm 1 C++SORT

```

Set values for threshold, maxAge and minHitStreak
for all frames do
  Get detections from frame
  Remove detections with confidence less than threshold
  Predict new locations of objects using predictors
  Associate predictors and detections based on similarity
  for all predictors do
    if predictor has associated detection then
      PROPAGATE(predictor, detection)
    else
      PROPAGATE(predictor)
    end if
  end for
  for all unmatchedDetections do
    Initialise new predictor with detection
  end for
  Remove predictors that have had no associated detection the last maxAge frames
  for all predictors do
    if predictor had associated detection this and the last minHitStreak frames then
      Add state of predictor to trackings
    end if
  end for
  return trackings
end for

```

Рисунок 3.1 – Код алгоритму C++SORT

Мережі разом з текстовими файлами для виявлень є загальнодоступними. Також, ми маємо доступ до основної інформації як тренувального так і тестового набору даних. Для деталізації роботи мереж включаючи інформацію, як вони тренують можна побачити у роботі Барекатейна [11].

3.1.2 Алгоритм передбачення

Досягнуто двох способів передбачення нових позицій, а саме фільтри Калмана та частотний фільтр, теорія яка представлена у розділі 2.2.2. Кожен алгоритм передбачування відстежує об'єкт, який моделює рух за допомогою фільтра. В додаток

до цього він продовжує відстежувати деякі метадані, такі як період видимості об'єкта або період неактивності алгоритму передбачення. Нажаль використання частотних фільтрів не є успішним через повільну швидкість роботи. Жоден обширний експеримент не був проведений з цим видом алгоритму передбачення.

Для будь-якого підходу положення x містить інформацію щодо центру піксельної координати, площі, співвідношення висоти і ширини, швидкість та зміну площі через деякий час. Спостереження z з іншого боку має значення, яке можна спостерігати лише у єдиному кадрі, а саме центр координати, площа та співвідношення ширини і висоти в обмежельному боксі. Площа та співвідношення утворюють пару, якій віддається перевага над збереженням ширини і висоти, так як співвідношення ширини і висоти припускається справедливо сталою протягом відеопослідовності. Це також є причиною чому зміна у співвідношенні через деякий час не моделюється у положенні x .

Отримуючи передбачене нове місце розташування об'єкту у кадрі $K+1$ прирівнюється до обчислення X_{K+1} і тоді зміна спостереження буде мати вигляд Z_{K+1} в рівняннях для динамічних систем. Рівняння (2.2), (2.3), (2.9) і (2.10). Однак усі змінні в спостереженні z містяться у положенні x , тому обчислення спостереження z виконується простим видаленням полів винятково до положення x . Так як ми не маємо контрольного сигналу u , обчислення для фільтру Калмана є $X_{K+1} = A x_K$. Обчислення для частотного фільтру буде трішки складнішим. Будова набору N частинок, кожна з яких має своє власне припущення правильного значення. Щоб отримати передбачене нове місцезнаходження, кожна частинка повинна збільшуватися на один крок, що є подібно до фільтру Калмана. Однак, після того, частинки потрібно скомбінувати у єдину міру. Для спрощення ми використовуємо середню вагу, але вона може також братися до уваги використовуючи вагу частинок, коли ми обчислюємо значення.

Вага яку ми використовуємо при повторному розміщенні частинок є IoU стану частинок та обмежувального боксу об'єкта відстеження Таким чином частинки які є близькими до об'єктів відстеження мають більшу вагу тоді як частинки які не

перетинаються з обмежувальним боксом – видаляються

Після об'єднання яке описано в наступному розділі прогнозування повинно пропагувати їхні штати до наступного кроку якщо передбачування об'єдналося з виявленням це описано в розділі 2.2.2 Інакше кажучи x_{k+1} обчислюється без регулювання будь-кого спостереження а також крок передбачення разом з фільтром Калмана При використанні частотного фільтра додається шум до поширення кожної частинки так як є зростаюча неточність щодо правильної позиції відстежувального об'єкту

Так як дуже примітивна основна лінія третя модель яка відноситься до стаціонарного алгоритму передбачення використовується у експериментах Okutama-Action як дуже примітивна основна лінія Ця назва означає що дана модель передбачає що всі об'єкти є нерухомими так як вона не моделює їх швидкість Інтуїтивно зрозуміло що цей алгоритм передбачення повинен більше потерпати від зменшення частоти кадрів ніж фільтр Калмана та частотний фільтр так як об'єкти подолають більшу відстань між кадрами

3.1.3 Об'єднання

Як описано у розділі 2.2.3 метою кроку об'єднання є знайти оптимальне співпадіння між алгоритмом передбачення та об'єктом виявлення якість співпадіння визначається сумою вартості кожної приєднаної пари алгоритму передбачення-об'єкт виявлення Залежачи від того чи висока ціна вказує на подібність або відмінність загальна сума повинна бути максимізована або мінімізована відповідно

Перший крок знаходження співпадіння є генеруванням ціни матриці Ця матриця де кожний елемент C_{ij} є схожістю або відмінністю між алгоритмом передбачення i та об'єктом виявлення j Задаючи ціну матриці угорський алгоритм може знайти оптимальне співпадіння між алгоритмом передбачення та об'єкту виявлення Однак перевагою є наявність порогового значення яке використовується щоб

відхилити об'єднання якщо ціна зависока або занижка Інакше якщо два послідовних кадри містять дві людини але якщо у першомуу кадрі виявлена лише одна людина а у другому інша то метод прийме два виявлення за одну і ту ж людину коли повинно бути очевидним що це не випадок отримання дистансії між двома людьми

Три ступені схожостей є поєднані та віднесені до IoU ціни лінійної ціни та експоненціальної ціни Помітно що усі ступені використовують лише геометричну та позиційні властивості обмежувальних боксів так як не використовується жодної інформації про вигляд або колір гістограм Рішення було прийняте на основі того що такі ступені сповільняють обробку обчислення або ж введуть небажані залежності на об'єкт виявлення якщо була використана з прихованих шарів

IoU ціна є рівною м IoU між алгоритмом прдбачення та виявлним обмежувальним боксом як наведено у рівнянні (2.15)

$$C_{IoU}(A, B) = \text{IoU}(A, B). \quad (3.1)$$

Для цього ступення високе значення вказує на схожість а поріг 0.3 було визначено показником хорошого результату Цей ступінь має зворотний бік де обмежувальний бокс без перетину дає нульове значення схожості незалежно від того як далеко вони знаходяться і таким чином не може бути відомо наскільки відмінними вони є ніж ті що не пересікаються Однак є переключенням той факт що використовується поріг тому всі неправильні співпадіння видаляються

Використовуючи формулу запропоновану Санчеса-Матілли [27] лінійна ціна є схожа до рівняння (2.16). Однак для реалізації причин ціна є перевернутою яка має вигляд

$$C_{lin}(A, B) = \sqrt{\frac{Q_{dist}}{(X_A - X_B)^2 + (Y_A - Y_B)^2}} \cdot \sqrt{\frac{Q_{shp}}{(H_A - H_B)^2 + (W_A - W_B)^2}}$$

З цією модифікацією цей ступінь ділить властивість інших двох модифікацій таким чином що високе значення вказує на схожість Було вивлено що 10000 це

хороше значення порогу Третій ступінь – експоненційна ціна є запропонованою Yu [26] у рівнянні 219 без використання зовнішніх ознак Таким чином ми отримуємо знову ж значення близьке до 1 вказує на схожість а 05 використовувалося як поріг для відкидання відмінних обмежувальних боксів.

3.2 MOTChallenge

Для мотчелендж є два доступних відстежувальних набори даних 2D MOT 2015 (який відтепер відноситься до MOT 15) та MOT 16 Оцінюючи тренувальний набір було вивлено що IoU ціна була найкращою для цього набору даних хоча експоненційна ціна давала схожі результати Невеликий мережевий пошук був застосований щоб знайти оптимальні значення для параметрів у алгоритмі 1 такі як пороги максимальний вік та мінімальна ударна смуга для алгоритму передбачення Пороги були раніше представлені у цьому розділі \maxAge був встановлений як 1 а \minHeatStreak як 3 Після вибору моделей та значення параметрів метод був тестований а вихідні дані були подані до мотчелендж Результати як і результати інших груп представлені у розділі 4.1 Помітно що це є та сама конфігурація яка була використана Бейвлі [22] Однак вони оцінили лише модель із звичайними виявленнями тим самим ускладнивши поєднання з іншою роботою так як якість виявлення може сильно впливати на виконання відстеження Більш того ми очікуємо зменшення часу виконання відповідно до вибору мови програмування (розділ 3.4).

3.3 Відстежування у різних кадрах

Другу частину цього проекту що заявлено на початку цього розділу розкаже нам про дослідження , про те як продуктивність трекера впливає на частоту кадрів вхідного відео. З тих пір відеопослідовності в мотчелендж є різноманітні в

роздільній здатності і частоті кадрів, які вирішували не в експериментах на базах даних, але замість їх перевірки на даних Okutama-Action. Тому що всі послідовності в Okutama-Action мають ту ж саму послідовність кадрів і загалом є більшу схожими один на одного, очікується що не буде кількох послідовностей на які сильно впливає зміна у послідовності кадрів. Тим самим викривляючи середню продуктивність. Більш того Okutama-Action має перевагу до доступу до двох різних детекторів об'єкту і їх виявлень, так як і до основних даних як тренувального так і тестувального набору.

Параметри встановлені з однаковими значеннями так як в мотчелендж. Це було виявлено при проведенні невеликої мепрежевої дослідницької роботи на тренувальному наборі. Кожна частина експерименту складалася з проведення докладної конфігурації на тестовий набір з сімома різними частотами кадрів: 30, 15, 10, 3, 2, 1. Так як відео було надано 30 fps, то при частоті кадрів 30 використовувалися усі кадри, в той час як частота кадрів в 1 означала, що лише кожен 30-й кадр буде використовуватися. Загалом використовувалося 18 різних конфігурацій(3 різних набори виявлень, 3 різних міри подібностей та 2 різних алгоритми передбачення). Виявлення з моделей виявлення людей, дія розпізнавання та основні дані для завдання виявлення об'єкту. Особливо для основних даних не використовувалося ніякої id інформації для того щоб отримати ідеальний результат виявлення об'єкту. Для заходів подібності, які раніше були зазначені – в IoU, функції лінійної та експоненціальної ціни. Так як хороших результатів не було отримано за допомогою частотних фільтрів лише фільтр Калмана та стаціонарні алгоритми передбачення використовуються. Головні результати були наведені у розділі 4.2, а таблиці усіх частин експерименту знаходяться у додатку А.

3.3.1 Виявлення та відстежування об'єктів у реальному часі

Як було сказано у розділі 2.3 частота кадрів в якому трекер може працювати у режимі реального часу впливає не лише на швидкість відстежування, але і на

швидкість об'єкта виявлення. Як приклад, повна модель – це виявлення об'єктів онлайн з використанням більшою мірою CNNs ніж завантаження виявлених текстових файлів та опрацювання на наборі даних Okutama-Action. Більш ніж використовували фіксовану частоту кадрів, кількість пропущених кадрів залежить від швидкості моделі, щоб впевнитись що немає комутативної затримки. В реальності, модель буде видавати останній записаний кадр так швидко, як тільки буде обчислений попередній кадр. В нашому випадку, ми вимірюємо час, який потрібно для моделі, щоб аналізувати кожний кадр та пропускаємо відповідну кількість кадрів. Нажаль, коли таким чином видаються кадри, ми не спроможні використовувати мотчелендж DevKit щоб виміряти дію, натомість вимірюється середня частота кадрів та використовуються результати іншого експерименту, щоб побачити чому відповідає значення. Щоб імітувати використовуваний випадок затримки повні моделі на вбудованій системі. Цей експеримент проводиться на Jetson TX1 Developer Kit, який завдяки своєму розміру може бути вмонтований в UAV щоб аналізувати охоплене відео. Результат цього експерименту показаний у розділі 4.2.1

3.4 Програмне середовище

Наша програма зроблена на C++ а не на Пайтоні через роботу Bewley [25]. Це є необхідним так як навколишнє середовище в яке буде інтегрований наш код написаний на C++ і тому написаний код іншої мови програмування призведе до небажаних результатів. Більш того C++ видає результати швидше в порівнянні з мовою програмування Пайтон.

Поглиблене вивчення моєї відбувалося у Caffe [50], який має оригінальну підтримку для C++ та GPU обчислення використовують CUDA. При спробі створити простіший код щоб розвинути наш новий механізм, лише 2 вихідні бібліотеки використовуються в додаток до Caffe [51] і Boost.

Для того щоб оцінити трекер використовувався мотчелендж який забезпечує

DevKit з скриптами Matlab. Вихідні дані трекара. Вихідні дані трекара збережені текстовими файлами, які є вхідними даними для скриптів Matlab. Вхідні дані порівнюються з основними даними, які також повинні бути введені та різні матриці обчислені та представлені.

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

4.1 MOTChallenge

Модель описана у розділі 3 працювала на наборі даних MOTChallenge, MOT15 та MOT16 за останні 2 роки. Різні конфігурації моделі працювали на тренувальному наборі, для виявлення найбільш ефективної моделі. Результати найкращих конфігурацій показані у таблицях 4.1 та 4.2 за відповідні 2 роки. Для обох наборів даних, використовуючи фільтр Калмана з експоненціальною та IoU вартістю був найбільш успішним. В додаток для MOT15 працювали ті ж конфігурації з основними даними як вхідними даними замість забезпечених виявлень. Як показано в таблиці це призвело до основного зростання дії, даючи майже ідеальне відстеження .

	MO TA	MO TP	F A F	M T	M L	F P	FN	ID Sw.	Fr ag
<i>Detections</i>									
IoU cost	25.8	72.5	1. 24	6 2	2 3 5	67 97	22 01 3	782	11 76
Exponential cost	23.1	72.4	1. 28	4 8	2 3 2	70 34	22 63 2	100 6	13 82
<i>Ground truth</i>									
IoU cost	95.8	94.8	0. 00	3 4 8	4 5	4	16 43	33	38
Exponential cost	96.0	94.7	0. 00	3 6 1	3 5	2	15 35	40	38

Таблиця 4.1 : результати використання фільтру Калмана з двома схожими мірами на тренувальному наборі MOT15. Вища частина це використання забезпечених виявлень, нижня частина це використання даних як вхідних даних.

Хоча через малу маржу, IoU вартість випереджала експоненціальн у вартість

(як мінімум при використанні нормальних виявлень). І тому була використана для тестового набору MOT15 та MOT16. Результати представлені у таблицях 4.3 та 4.4. Також в цих таблицях є результати інших груп взятих з таблиць лідерів онлайн [10]. Лише групи, які використовують невизначені виявлення і чий методи є в режимі онлайн (є протилежністю до офлайн див.Розділ 2.1) показані у таблиці

	MO TA	MO TP	F A F	M T	M L	FP	FN	ID Sw.	Fra g
<i>Detections</i>									
IoU cost	27.4	78.8	0. 5 9	2 2	3 0 2	31 30	76 20 2	786	11 65
Exponential cost	27.3	78.8	0. 6 1	2 2	3 0 3	32 34	76 13 7	837	12 40

Таблиця 4.2 : результати використання фільтрів Калмана з двома різними заходами подібності на тренувальному наборі MOT16. Представлені виявлення використовуються як вхідні дані.

Так як порівняння з іншими методами не є основним у нашому випадку.В додаток представлені лише результати груп, які зробили їх код доступним або описали їх роботу в доповіді (без анонімних посилань на MOTChallenge). Це є зроблено, так як немає способу підтвердити або аналізувати результати анонімних груп. Методи, які є впорядковані за їх середнім порядком через усі матриці, які представлені у лідируючих таблицях. Це робиться з метою ненаголошувати на єдиній матриці.

	M OT A	M OT P	F A F	M T	M L	F P	F N	ID Sw .	F r a g	H z
TDAM [52]	33. 0	72. 8	1 .	1 3.	3 9.	10 06	30 61	46 4	1 5	5. 9

MDPNN [53]	37.6	71.7	1.4	15.8	26.8	79.33	2939.7	10.26	0.62024	1.9
SCEA [54]	29.1	71.1	1.0	8.9	4.73	60.60	36.912	60.4	1.182	6.8
CDA_DDAL pb [55]	32.8	70.7	0.9	9.7	4.22	4983	35.690	61.4	1.583	2.3
MDP [34]	30.3	71.3	1.7	1.30	3.84	97.17	32.422	68.0	1.500	1.1
oICF [56]	27.1	70.0	1.3	6.4	4.87	75.94	36.757	45.4	1.660	1.4
LDCT [57]	4.7	71.7	2.4	1.14	3.25	14.066	32.156	12.348	2.2918	20.7
OMT_DFH [58]	21.2	69.9	2.3	7.1	4.65	13.218	34.657	56.3	1.255	28.6
GMPHD_15 [59]	18.5	70.9	1.4	3.9	5.53	78.64	41.766	459	1.266	19.8
EAMTTpub [27]	22.3	70.8	1.4	5.4	5.27	79.24	38.982	83.3	1.485	12.2
GSCR [60]	15.8	69.4	1.3	1.8	6.10	75.97	43.633	51.4	101	28.1
C++SORT	21.7	71.2	1.5	3.7	4.91	84.22	38.454	12.31	2.2005	1112.1
TSDA_OAL [61]	18.6	69.7	2.8	9.4	4.23	16.350	32.853	80.6	1.544	19.7
RNN_LSTM [35]	19.0	71.0	2.0	5.5	4.56	11.578	36.706	14.90	2.208	16.5.2
RMOT [62]	18.6	69.6	2.2	5.3	5.33	12.473	36.835	68.4	1.282	7.9
TC_ODAL	15.	70.	2.	3.	5.	12.	38.	63.	1.	1.

[63]	1	5	.	.	5.	97	53	7	7	7
			2	2	8	0	8		1	
									6	

Таблиця 4.3 : Лідируюча таблиця тестового набору MOT15 взята із MOTChallenge.

Найкращі дані в кожній категорії є виділені жирним шрифтом. Метод цієї доповіді, C++ SORT виділений сірим кольором .

	M OT A	M OT P	F A F	M T	M L	F P	FN	ID Sw .	F r a g e	H z
MDPNN16 [53]	47. 2	75. 8	0 5	1 4. 0	4 1 6	26 81	92 85 6	77 4	1 6 7 5	1. 0
oICF [56]	43. 2	74. 3	1 1	1 1. 3	4 8 5	66 51	96 51 5	38 1	1 4 0 4	0. 4
GMPHP_HD A [59]	30. 5	75. 4	0 9	4 6	5 9 7	51 69	12 09 70	53 9	7 3 1	13 .6
CDA_DDAL v2 [55]	43. 9	74. 7	1 .	1 0. 7	4 4 .	64 50	95 17 5	67 6	1 7 9	0. 5
C++SORT	31. 5	77. 3	0 5	4 3	5 9 9	30 48	12 02 78	15 87	2 2 3 9	68 7. 1
EAMTT_pub [27]	38. 8	75. 1	1 4	7 9	4 9 1	81 14	10 24 52	96 5	1 6 5 7	11 .8
OVBT [64]	38. 4	75. 4	1 .	7 5	4 7 3	11 51 7	99 46 3	13 21	2 1 4	0. 3

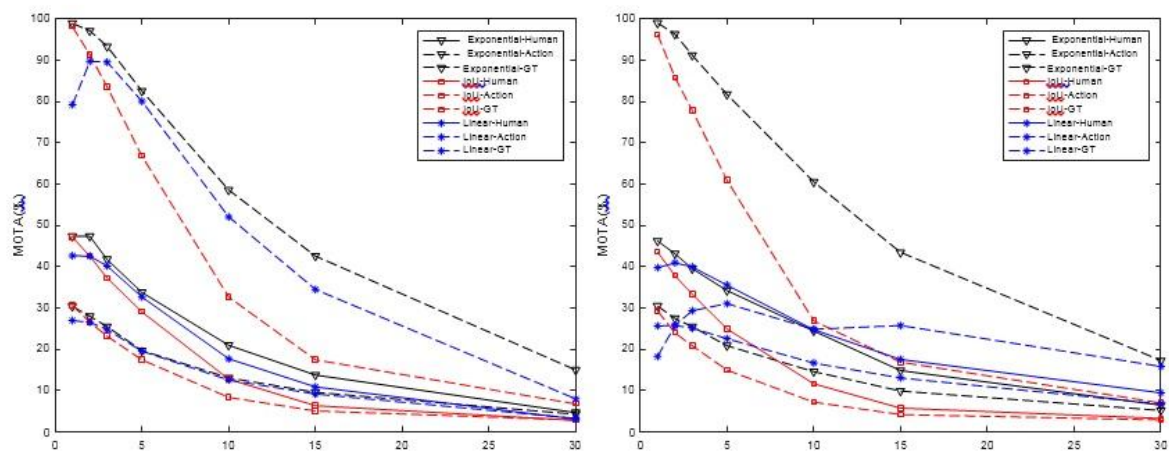
Таблиця 4.4 Лідируюча таблиця тестового набору MOT16 взята із MOTChallenge.

Наша робота називається C++ SORT і вона виділена сірим кольором.

Найкращий показник в матриці виділений жирним шрифтом. В обох експериментах наша робота значною мірою випереджає конкурентів у швидкості (Hz). Так от у MOT16 є найкращий метод в умовах MOTP і FAF. Однак в той же рік він зайняв останнє місце під час вимірювання кількості майже відстежених (MT) та майже втрачених (ML) траєкторій так як і кількість ID перемикачів та фрагментацій. Однак у MOT16 наша робота суперечливо зупинилася посередині не маючи найгіршої матриці і також не виграючи в термінах швидкості. Як заключна частина SORT [25] для якого, результати забезпечених виявлень недоступні, працює при 260.5 fps на MOT15, вказуючи на те що вибір мови програмування може великою мірою впливати на швидкість.

4.2 Відстеження з різною частотою кадрів

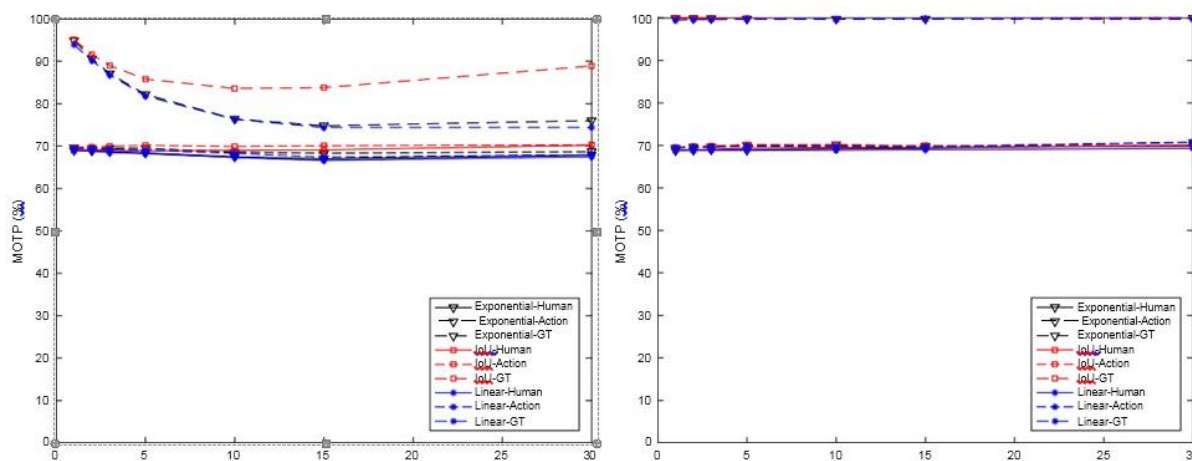
Наша модель працювала на тестовому наборі Okutama-Action використовуючи 18 тестових наборів конфігурацій, кожна з яких працювала при відборі проб відео з частотою в різних сім інтервалів. Відбір проб інтервалів і що кожний i -тий фрейм використовується і відповідає частоті кадрів $1/i$ FPS. Дія в межах MOTA і MOTP та відсоток траєкторії не позначений як майже втрачений. Для різної частоти кадрів як показано на рисунках 4.1 та 4.3. Кожен план містить 9 конфігурацій : три заходи подібності – експоненціальна, IoU і лінійної і 3 різних набори виявлення – від моделі виявлення людини (Людина), модель розпізнавання дій (Дія) та основні дані (GT). Заходи подібності є досягнутими за допомогою кольору та стилю маркера в той час як виявлення мають різні види ліній. Числові значення усіх матриць забезпечені MOTChallenge DevKit можна знайти в додатку A.



(а) використання фільтрів Калмана щоб передбачити нові локації. (б) стаціонарні передбачення нових локацій.

Рисунок 4.1 – МОТА для різних заходів подібності – це експоненціальна вартість в чорному трикутнику, вартість IoU в червоному квадраті, лінійна вартість в синій зірці, та різні типи виявлень – виявлення людини в суцільній лінії, розпізнавання дії показані штрихпунктирною лінією, а основні дані пунктирною лінією – показано як функції за інтервал вибірки.

В загальному для усіх матриць трекер працює краще з більш чіткими виявленнями. Також слід зазначити, що МОТА та відсоток відстежених траєкторій стрімко падає у зменшеній частоті кадрів. Загалом використання фільтрів Калмана прискорює дію злегка порівнюючи її з постійним алгоритмом передбачення хоча і не зовсім послідовно. Рисунок 4.2 показує, що МОТР почав зростати, коли частота кадрів була достатньо низькою, а також те, що постійні алгоритми передбачення можуть перевершити використання фільтрів Калмана. Під час використання алгоритмів передбачення та основних даних виявлень як вхідних даних, усі заходи подібності мали МОТП близьких до ста відсотків у будь-якій частоті кадрів.



(а) використання фільтрів Калмана щоб передбачити нові локації. (б)стаціонарні передбачення нових локацій.

Рисунок 4.2 – МОТП для різних заходів подібності – це експоненціальна вартість в чорному трикутнику, вартість IoU в червоному квадраті, лінійна вартість в синій зірці, та різні типи виявлень – виявлення людини в суцільній лінії, розпізнавання дії показані штрихпунктирною лінією, а основні дані пунктирною лінією – показано як функції за інтервал вибірки.

(а) використання фільтрів Калмана щоб передбачити нові локації

(б)стаціонарні передбачення нових локацій.их

Відсоток траєкторій, які ми не вважаємо майже втраченими (ML), для різних для різних заходів подібності – це експоненціальна вартість в чорному трикутнику, вартість IoU в червоному квадраті, лінійна вартість в синій зірці, та різні типи виявлень – виявлення людини в суцільній лінії, розпізнавання дії показані штрихпунктирною лінією, а основні дані пунктирною лінією – показано як функції за інтервал вибірки.

4.2.1 Відстеження в реальному часі

Опрацьовуючи повну модель на Jetson Developer Kit він може працювати на 4

FPS, використовуючи модель відстеження людини, і у 2 FPS використовуючи модель розпізнавання дій. Це відповідає пробному інтервалу 7.5 і відповідно 15. Як може бути видно на рисунках 4.1 та 4.2 робота з такою частотою кадрів є значно нижча ніж при використанні усіх кадрів. Ці рисунки висвітлюють МОТА в найкращих конфігураціях при пробних інтервалах відповідних роботі в реальному часі. Якщо порівняти дію аналізуючи усі кадри, то робота C++ SORT в реальному часі з моделлю людського виявлення призводить до падіння ефективності майже до 50 %. Коли використовуємо модель розпізнавання дій ефективність знижується на третину в порівнянні з використанням усіх кадрів.

Рисунок показує, що МОТА при використанні фільтру Калмана з використанням різних заходів подібності – експоненціальна вартість в чорному трикутнику, вартість IoU в червоному квадраті, лінійна вартість в синій зірці, та різні типи виявлень – виявлення людини в суцільній лінії, розпізнавання дії показані штрихпунктирною лінією, а основні дані пунктирною лінією – показано як функції за інтервал вибірки. Кути, які позначені пунктирною лінією та точковою лінією показують втрату ефективності при роботі на C++ SORT – з моделями виявлення людини та розпізнавання дій в реальному часі.

5 ЕКОНОМІЧНИЙ РОЗДІЛ

5.1 Технологічний аудит розробленої системи відеоспостереження на базі рухомого об'єкта

Як було зазначено вище, питання відстеження переміщення різних рухових об'єктів з кожним днем стає все більш актуальним. Практична реалізація поставленого завдання потребує проведення відповідних досліджень, а саме: вирішення завдання миттєвої обробки зображень при відстеженні переміщення об'єкта під час навігації за даними відеокамер спостереження.

Слід підкреслити, що сфери зображень та відеоаналізу за останні роки привернули увагу багатьох дослідників частково завдяки появі згорткових нейронних мереж (CNNs), використання яких показало хороші результати.

Наявність точних моделей зображень та відеоаналізу стали основою для автономних транспортних засобів та роботів, дали можливість самостійно керувати автомобілями, безпілотними літальними апаратами тощо.

Сьогодні не виникає сумнівів, що сучасні системи відеоспостереження відіграють важливу роль у підвищенні рівня безпеки та дотримання прав приватної власності об'єкта, який перебуває під охороною. Щодня компанії, що працюють у сфері систем охорони, розробляють нові технології, які здатні ефективно усувати несприятливі ситуації та підвищувати рівень безпеки. Незважаючи на те, що традиційні системи відеоспостереження є досить поширеними та демонструють високий рівень захисту, вони мають певні недоліки, основним з яких є високі затрати пам'яті на збереження зафіксованого відеоматеріалу, що призводить до збільшення часу, який витрачається на перегляд та аналіз збережених матеріалів.

З метою усунення цього недоліку і підвищення ефективності функціонування існуючих систем відеоспостереження і були проведені ці дослідження. Перед нами було поставлено завдання – зменшити розмір пам'яті, яка витрачається на збереження відеоматеріалів, а також розробити ефективні програмні засоби розпізнавання руху

об'єкта для інтеграції у системи відеоспостереження.

В результаті виконаної магістерської роботи було розроблено систему відеоспостереження, яка дає можливість значно зменшити розмір пам'яті, відведений для збереження відеоматеріалів. Крім того ця система може бути інтегрована в сучасні системи відеоспостереження майже без апаратних змін.

Для встановлення технічного рівня та комерційного потенціалу отриманих в ході досліджень результатів проведемо їх технологічний аудит. Для проведення аудиту були запрошені три експерти, які є фахівцями з цього питання: к.т.н. Гармаш В.В., Осіковський П.О. та Карпенко В.В.

При проведенні технологічного аудиту експерти користувалися рекомендаціями Державного комітету України з питань науки, інновацій та інформатики [1], наведеними в таблиці 4.1.

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено робоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри тері й	0	1	2	3	4
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Таблиця 5.1 – Критерії для проведення технологічного аудиту розробки та їх бальна оцінка

Результати проведеного технологічного аудиту зведено в таблицю 5.2.

Критерії	Прізвище, ініціали експерта		
	Гармаш В.В.	Осіковський П.О.	Карпенко В.В.
	Бали, виставлені експертами:		
1	2	2	2
2	3	2	2
3	3	2	3
4	3	3	3
5	3	3	2
6	3	3	3
7	3	3	2
8	2	2	3
9	2	2	2
10	2	2	3
11	2	2	2
12	2	2	2

Сума балів	СБ ₁ =30	СБ ₂ = 28	СБ ₃ = 29
Таблиця 5.2 – Результати технологічного аудиту нашої розробки			
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{30 + 28 + 29}{3} = \frac{87}{3} = 29$		

На підставі рекомендацій, наведених в [1] (див. табл. 4.3), можна зробити висновок, що розроблена нами система відеоспостереження на базі рухомого об'єкта має технічний рівень та комерційний потенціал, який характеризується як «середній».

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Технічний рівень та комерційний потенціал розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Таблиця 5.3 – Технічні рівні та комерційний потенціал розробки

Такий рівень нашої розробки пояснюється тим, що на основі проведеного порівняльного аналізу роботи звичайної системи відеоспостереження та системи, яка використовує метод розпізнавання рухомих об'єктів, зроблено рекомендації щодо того, в яких ситуаціях наша система відеоспостереження буде корисною та встановлено її переваги і недоліки.

5.2 Розрахунок витрат на проведення досліджень

Для визначення витрат на проведення наукових досліджень та розробку системи відеоспостереження на базі рухомого об'єкта розрахуємо такі основні статті [2]:

а). Основна заробітна плата Z_o дослідників:

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн,} \quad (4.1)$$

де M – місячний посадовий оклад конкретного виконавця, грн. Величини окладів знаходиться в межах (4173...19000) грн. за місяць; T_p – число робочих днів в місяці; прийmemo $T_p = 24$ дні; t – число робочих днів роботи виконавців роботи.

Зроблені розрахунки зведемо до таблиці 4.4:

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів (або годин) роботи	Витрати на оплату праці, грн	Примітка
1. Керівник магістерської роботи	10800	450	25 годин	1875	Робочий день – 6 годин
2. Студент-магістрант	2150	89,58	52	4658,16	
3. Консультант з ЕЧ	11040	460	2,5 годин	191,67	Робочий день – 6 годин
4. Технічні працівники	7200	300	2	600	
Всього				$Z_o=7324,83 \approx 7325$ грн	

Таблиця 5.4 – Основна заробітна плата виконавців роботи

б). Додаткова заробітна плата Z_d виконавців:

$$Z_d = (0,1...0,12) \cdot Z_o. \quad (4.2)$$

Для нашого випадку будемо мати:

$$Z_d = 0,105 \times 7325 = 769,13 \approx 770 \text{ грн.}$$

в). Нарахування на заробітну плату $N_{зп}$ виконавців:

$$H_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100}, \quad (4.3)$$

де β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %. $\beta = 22\%$. Тоді:

$$H_{зп} = (7325 + 770) \times 0,22 = 1780,90 \approx 1781 \text{ грн.}$$

г). Амортизація A обладнання, комп'ютерів та приміщень розраховується за формулою :

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ грн,} \quad (4.4)$$

де $Ц$ – загальна балансова вартість основних засобів, які використовувалися;

H_a – річна норма амортизаційних відрахувань. Спрощено можна прийняти, що $H_a = (2,5...25)\%$; T – термін, використання основних засобів, місяці.

Зроблені розрахунки зведемо у таблицю 4.5.

Найменування основних засобів	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
1. Обладнання, відеокамера спостереження, комп'ютер, принтер тощо	22500	20	2,5 (при 50% викорис-танні)	468,75
2. Приміщення кафедри та факультету	8500	4	3 (при 50% викорис-танні)	42,5 $A = 512$
Всього				$A = 512$

Таблиця 5.5 – Розрахунок амортизаційних відрахувань

д). Витрати на матеріали M розраховуються за формулою:

$$M = \sum_1^n H_i \cdot C_i \cdot K_i - \sum_1^n B_i \cdot C_B \text{ грн.}, \quad (4.5)$$

де H_i – витрати матеріалу i -го найменування, кг; C_i – вартість матеріалу i -го найменування, грн./кг; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; B_i – маса відходів матеріалу i -го найменування, кг; C_B – ціна відходів матеріалу i -го найменування, грн/кг; n – кількість видів матеріалів.

Витрати на комплектуючі K розраховуються за формулою:

$$K = \sum_1^n H_i \cdot C_i \cdot K_i \text{ грн.}, \quad (4.6)$$

де H_i – кількість комплектуючих i -го виду, шт.; C_i – ціна комплектуючих i -го виду, грн; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; n – кількість видів комплектуючих.

Загальна вартість основних матеріалів та комплектуючих, які були використані під час проведення досліджень та розробки системи відеоспостереження на базі рухомого об'єкта, складає приблизно 2000 грн.

е). Витрати на силову електроенергію V_e розраховуються за формулою:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d}, \quad (4.7)$$

де V – вартість 1 кВт-год. електроенергії, в 2019 р. $V \approx 2,2$ грн/кВт; Π – установлена потужність обладнання, кВт; $\Pi = 0,88$ кВт; Φ – фактична кількість годин роботи обладнання, годин.

Прийmemo, що $\Phi = 150$ годин; $K_{\text{п}}$ – коефіцієнт використання потужності; $K_{\text{п}} < 1 = 0,8$; $K_{\text{д}}$ – коефіцієнт корисної дії, $K_{\text{д}} = 0,6$.

Тоді витрати на електроенергію складуть:

$$V_e = \frac{B \cdot \Pi \cdot \Phi \cdot K_{\text{п}}}{K_{\text{д}}} = \frac{2,2 \cdot 0,88 \cdot 150 \cdot 0,8}{0,6} = 387,20 \approx 388 \text{ грн.}$$

ж). Інші витрати $V_{\text{ін}}$ можна прийняти як (100...300)% від суми основної заробітної плати виконавців, тобто:

$$V_{\text{ін}} = (1..3) \times Z_0. \quad (4.8)$$

Для нашого випадку: $V_{\text{ін}} = 2,0 \times 7325 = 14650$ грн.

и). Сума всіх попередніх статей витрат дає витрати на виконання цього етапу роботи – V .

$$V = 7325 + 770 + 1781 + 512 + 2000 + 388 + 14650 = 27426 \text{ грн.}$$

к) Загальні витрати ZB на проведення подальших досліджень та завершення роботи розраховуються за формулою:

$$ZB = \frac{V}{\beta}, \quad (4.9)$$

де β – коефіцієнт, який характеризує етап виконання цієї роботи на шляху до її можливого впровадження. Якщо розробка знаходиться: на стадії науково-дослідних робіт, то $\beta \approx 0,1$; на стадії технічного проектування, то $\beta \approx 0,2$; на стадії розробки конструкторської документації, то $\beta \approx 0,3$; на стадії розробки технологій, то $\beta \approx 0,4$; на стадії розробки дослідного зразка, то $\beta \approx 0,5$; на стадії розробки

промислового зразка, $\beta \approx 0,7$; на стадії впровадження, то $\beta \approx 0,8-0,9$.

Оскільки наша розробка потребує ще певного доопрацювання, то для нашого випадку доцільно прийняти, що $\beta \approx 0,5$.

Тоді:

$$ЗВ = \frac{27426}{0,5} = 54852 \text{ грн.}$$

Тобто прогнозовані витрати на проведення подальших досліджень та завершення нашої розробки становлять приблизно 55 тис. грн.

5.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки

Немає сумнівів, що розроблена нами система відеоспостереження буде користуватися значним попитом на ринку, оскільки вона призначена для охо-рони територій та приміщень, що в сучасних нестабільних економічних умовах і зростання злочинності є край необхідним.

Аналіз місткості ринку аналогічної продукції, проведений нами, показав, що на цей час в Україні кількість реальних та потенційних користувачів подібних розробок складає щороку приблизно 100 господарюючих суб'єктів та інших осіб і їх кількість буде постійно зростати. Оскільки розроблена нами система відеоспостереження має значно кращі функціональні можливості та вищу ефективність, то її можна буде реалізовувати на ринку дещо дорожче, ніж системи-аналоги. Якщо середня ціна подібних систем становить сьогодні 12 тис. грн, то нашу систему відеоспостереження можна буде продавати приблизно за 15 тис. грн, тобто на 3 тис. грн дорожче.

Розроблену нами систему відеоспостереження можна буде реалізовувати на ринку з 1 січня 2021 року (оскільки вона потребує ще деякого доопрацювання), а її результати будуть виявлятися протягом 2021-го, 2022-го та 2023-го років. Прогноз зростання попиту на нашу розробку складає по роках: 1-й рік після впровадження

(2021 р.) – приблизно +30 шт.; 2-й рік після впровадження (2022 р.) – приблизно +45 шт.; 3-й рік після впровадження (2023 р.) – приблизно +60 шт.

Можливе збільшення чистого прибутку $\Delta\Pi_i$, що його може отримати потенційний інвестор від фінансування та впровадження розробленої нами системи відеоспостереження становитиме [2]:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100}\right), \quad (4.10)$$

де $\Delta\Pi_o$ – покращення основного якісного показника від впровадження результатів розробки у цьому році. Таким показником є збільшення ціни нової розробки, грн. Для нашого випадку це буде $\Delta\Pi_o = 15 - 12 = 5$ тис. грн; N – основний кількісний показник, який визначає обсяг діяльності у цьому році до впровадження результатів розробки; $N = 100$ шт.; ΔN – покращення основного кількісного показника від впровадження результатів розробки. Таке покращення по роках становитиме, відповідно: +30, +45 та +60 шт.; Π_o – основний якісний показник, який визначає обсяг діяльності (тобто ціну) у році після впровадження розробки, грн; $\Pi_o = 15$ тис. грн; n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; $n = 3$ роки; λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = (0,2 \dots 0,5)$; візьмемо $\rho = 0,5$; υ – ставка податку на прибуток. У 2019 році $\upsilon = 18\%$.

Величина чистого прибутку $\Delta\Pi_1$ для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2021 р.) складе:

$$\Delta\Pi_1 = [3 \cdot 100 + 15 \cdot 30] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) = 256,24 \approx 257 \text{ тис. грн.}$$

Величина чистого прибутку $\Delta\Pi_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2022 р.) року складе:

$$\Delta\Pi_2 = [3 \cdot 100 + 15 \cdot 45] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) = 333,12 \approx 334 \text{ тис. грн.}$$

Величина чистого прибутку $\Delta\Pi_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2023 р.) року складе:

$$\Delta\Pi_3 = [3 \cdot 100 + 15 \cdot 60] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) = 409,98 \approx 410 \text{ тис. грн.}$$

Теперішня вартість інвестицій PV , що мають бути вкладені в нашу розробку:

$$PV = (2 \dots 5) \times 3B.$$

Для нашого випадку $PV = 2,4 \times 55 = 132$ тис. грн.

Абсолютний ефект вкладених інвестицій $E_{\text{абс}}$ може становити:

$$E_{\text{абс}} = \text{ПП} - PV, \quad (4.11)$$

де ПП – приведена вартість всіх можливих чистих прибутків від можливого впровадження нашої розробки, грн; PV – теперішня вартість інвестицій $PV = 132$ тис. грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$\text{ПП} = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.12)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн; t – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку $t = 3$ роки; τ – ставка дисконтування. Керуючись даними статистики за 2019 рік, приймемо ставку дисконтування $\tau = 0,05$ (5%); t – період часу від моменту

отримання прибутків до початку впровадження розробки.

Тоді приведена вартість всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження нашої розробки, складе:

$$\text{ПП} = \frac{257}{(1+0,05)^2} + \frac{334}{(1+0,05)^3} + \frac{410}{(1+0,05)^4} \approx 233 + 289 + 337 = 859 \text{ тис. грн.}$$

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$$E_{\text{абс}} = 859 - 132 = 727 \text{ тис. грн. або } 242,33 \text{ тис. грн щорічно.}$$

Далі розрахуємо внутрішню норму дохідності E_B вкладених коштів:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (4.13)$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених коштів; $E_{\text{абс}} = 859$ тис. грн; PV –теперішня вартість початкових інвестицій $PV = 132$ тис. грн; $T_{\text{ж}}$ – життєвий цикл розробки, роки. $T_{\text{ж}} = 4$ роки.

Для нашого випадку будемо мати:

$$E_B = \sqrt[4]{1 + \frac{859}{132}} - 1 = \sqrt[4]{1 + 6,5075} - 1 = \sqrt[4]{7,5075} - 1 = 1,655 - 1 = 0,655 = 65,5\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційний інвестор не буде вкладати кошти в нашу розробку. Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f \quad (4.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = (0,12...0,15)$; f –показник, що характер

Для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,15 + 0,4 = 0,55 \text{ або } \tau_{\text{мін}} = 55\%.$$

Оскільки величина $E_B = 65,5\% > \tau_{\text{мін}} = 55\%$, то потенційний інвестор може бути зацікавлений у фінансуванні нашої розробки.

Далі розраховуємо термін окупності коштів, вкладених у розробку та впровадження нашої системи відеоспостереження. Термін окупності $T_{\text{ок}}$ можна розрахувати за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}. \quad (4.15)$$

Для нашого випадку термін окупності $T_{\text{ок}}$ коштів складе:

$$T_{\text{ок}} = \frac{1}{0,655} \approx 1,5267 \text{ років,}$$

що свідчить про потенційну доцільність комерціалізації нашої розробки.

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю:

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку системи відеоспостереження	Не більше 60 тис. грн	55 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження системи відеоспостереження, тис. грн	не менше 200 тис. грн за рік	242,33 тис. грн щороку	Виконано

3. Внутрішня норма дохідності інвестицій, %	не менше 50%	65,5%	Досягнуто
4. Термін окупності, роки	до 3-х років	1,5267 років	Виконано

Таким чином, основні техніко-економічні показники розробленої нами системи відеоспостереження на базі рухомого об'єкта, визначені у технічному завданні, виконані.

ВИСНОВКИ

У моїй магістерській роботі реалізовано трекер, як розширення існуючого методу слідуючи парадигмі відстеження за виявленням. Він працює за допомогою моделювання руху об'єктів шляхом вирішення проблеми фільтрації та асоціювання виявлення з передбачуваними новими місцями в нових кадрах за допомогою угорського алгоритму. Використовуються три різні подібності, які використовують розташування та форму обмежувальних боксів. Порівняно з іншими трекерами в таблиці MOTChallenge, наш метод, який називається C ++ SORT, є найшвидшим поданням, а також досягає гідних балів за іншими показниками. Запускаючи нашу модель на наборі даних вибірки з різною швидкістю кадрів, ми показуємо, що продуктивність значно знижується при запуску моделі включаючи виявлення об'єктів у режимі реального часу. У більшості показників оцінка знижується на 50%, але в деяких випадках на 90%. Це вказує на те, що інші, повільніші методи не можуть бути використані для відстеження в режимі реального часу, але для цього потрібно конкретніше дослідження.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
3. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.
4. Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
5. Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
6. Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108*, 2015.
7. NASA. Unmanned aircraft system (UAS) traffic management (UTM), 2017. URL <https://utm.arc.nasa.gov/>. Accessed: May 24, 2017.
8. Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*, 2016.
9. MOTChallenge leaderboard, 2016. URL <https://motchallenge.net/results/MOT16/>. Accessed: May 24, 2017.
10. Mohammadamin Barekatin, Miquel Martí, Hsueh-Fu Shih, Samuel Murray, Kotaro Nakayama, Yutaka Matsuo, and Helmut Prendinger. Okutama-Action: An aerial

view video dataset for concurrent human action detection. In *1st Joint BMTT-PETS Workshop on Tracking and Surveillance, CVPR*, pages 1–8. IEEE, 2017.

11. Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, 2014.

12. Hao Jiang, Sidney Fels, and James J Little. A linear programming approach for multiple object tracking. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8. IEEE, 2007.

13. Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011.

14. Jerome Berclaz, Francois Fleuret, and Pascal Fua. Multiple object tracking using flow linear programming. In *Performance Evaluation of Tracking and Surveillance (PETS- Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–8. IEEE, 2009.

15. Byungjae Lee, Enkhbayar Erdenee, Songguo Jin, Mi Young Nam, Young Giu Jung, and Phill Kyu Rhee. Multi-class multi-object tracking using changing point detection. In *European Conference on Computer Vision*, pages 68–83. Springer, 2016.

16. Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

17. Massimo Piccardi. Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on*, volume 4, pages 3099–3104. IEEE, 2004.

18. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.

19. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

20. ImageNet leaderboard, 2016. URL <http://image-net.org/challenges/>

LSVRC/2016/results. Accessed: May 24, 2017.

21. PASCAL VOC leaderboard. URL <http://host.robots.ox.ac.uk:8080/leaderboard/>. Accessed: May 24, 2017.
22. COCO leaderboard. URL <http://mscoco.org/dataset/#detections-leaderboard>. Accessed: May 24, 2017.
23. Shai Avidan. Support vector tracking. *IEEE transactions on pattern analysis and machine intelligence*, 26(8):1064–1072, 2004.
24. Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple on-line and realtime tracking. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3464–3468. IEEE, 2016.
25. Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016.
26. Ricardo Sanchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision*, pages 84–99. Springer, 2016.
27. Chanh Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.
28. T Klinger, F Rottensteiner, and C Heipke. Probabilistic multi-person tracking using dynamic bayes networks. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2015.
29. L. D. Jackel. Handwritten digit recognition with a back-propagation network. In David Touretzky, editor, *Advances in Neural Information Processing Systems (NIPS 1989)*, volume 2, Denver, CO, 1990. Morgan Kaufman.
30. Yi Li, Kaiming He, Jian Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.
31. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards

real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

32. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng- Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

33. Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4705–4713, 2015.

34. Anton Milan, Seyed Hamid Rezaatofghi, Anthony Dick, Ian Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. *arXiv preprint arXiv:1604.03635*, 2016.

35. Peter Ondruska and Ingmar Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. *arXiv preprint arXiv:1602.00991*, 2016.

36. Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *arXiv preprint arXiv:1703.07402*, 2017.

37. Kevin Smith, Daniel Gatica-Perez, and J-M Odobez. Using particles to track varying numbers of interacting people. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 962–969. IEEE, 2005.

38. Rudolph Emil Kalman. A new approach to linear filtering and prediction problems.

39. David Salmond and Neil Gordon. An introduction to particle filters. *State space and unobserved component models theory and applications*, pages 1–19, 2005.

40. Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

41. Jack Edmonds and Richard M Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

42. Cheng-Hao Kuo, Chang Huang, and Ramakant Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 685–692. IEEE, 2010.

43. Sen M Kuo, Bob H Lee, and Wenshun Tian. *Real-time digital signal processing: fundamentals, implementations and applications*. John Wiley & Sons, 2013.
44. Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.
45. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, March 2016. URL <http://arxiv.org/abs/1603.00831>. arXiv: 1603.00831.
46. Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):1–10, 2008.
47. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
48. Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.
49. Min Yang and Yunde Jia. Temporal dynamic appearance modeling for online multi-person tracking. *Computer Vision and Image Understanding*, 153:16–28, 2016.
50. Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 2017.
51. Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. Online multi-object tracking via structural constraint event aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1392–1400, 2016.
52. Seung-Hwan Bae and Kuk-Jin Yoon. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
53. Hilke Kieritz, Stefan Becker, Wolfgang Hübner, and Michael Arens. Online multi-person tracking using integral channel features. In *Advanced Video and Signal Based*

Surveillance (AVSS), 2016 13th IEEE International Conference on, pages 122–130. IEEE, 2016.

54. Francesco Solera, Simone Calderara, and Rita Cucchiara. Learning to divide and conquer for online multi-target tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4373–4381, 2015.

55. Jaeyong Ju, Daehun Kim, Bonhwa Ku, David K Han, and Hanseok Ko. Online multi-object tracking with efficient track drift and fragmentation handling. *JOSA A*, 34(2):280–293, 2017.

56. Young-min Song and Moongu Jeon. Online multiple object tracking with the hierarchically adopted gm-phd filter using motion and appearance. In *Consumer Electronics-Asia (ICCE-Asia), IEEE International Conference on*, pages 1–4. IEEE, 2016.

57. Low Fagot-Bouquet, Romaric Audigier, Yoann Dhome, and Frédéric Lerasle. Online multi-person tracking based on global sparse collaborative representations. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2414–2418. IEEE, 2015.

58. Jaeyong Ju, Daehun Kim, Bonhwa Ku, David K Han, and Hanseok Ko. Online multi-person tracking with two-stage data association and online appearance model learning. *IET Computer Vision*, 2016.

59. Ju Hong Yoon, Ming-Hsuan Yang, Jongwoo Lim, and Kuk-Jin Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 33–40. IEEE, 2015.

60. Seung-Hwan Bae and Kuk-Jin Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1218–1225, 2014.

61. Yutong Ban, Sileye Ba, Xavier Alameda-Pineda, and Radu Horaud. Tracking multiple persons based on a variational bayesian model. In *European Conference on Computer Vision*, pages 52–67. Springer, 2016.

62. Відеоспостереження і охоронні системи [Електронний ресурс]. – Режим доступу: <http://www.install.in.ua>. Назва з екрану.

63. Відеоспостереження, спеціалізована відео техніка [Електронний ресурс]. – Режим доступу : <http://kaschtan.com.ua/shop> – Назва з екрану.
64. Системи відеоспостереження [Електронний ресурс]. – Режим доступу: <http://www.bastion.kiev.ua/Main/Videonablyudenie> – Назва з екрану.
65. Все про відеокамери в Україні [Електронний ресурс]. – Режим доступу: <http://videokamera.in.ua> – Назва з екрану.
66. Системи відеоспостереження для чайників [Електронний ресурс]. – Режим доступу: <http://www.sec4all.net/cameras1.html> – Назва з екрану.
67. Відеокамери види та різниця [Електронний ресурс]. – Режим доступу: URL:<http://www.bezpekacity.com.ua7> – Назва з екрану.
68. Надійність, діагностика та експлуатація комп'ютерних систем та мереж [Електронний ресурс]. – Режим доступу: <http://dtgt.dp.ua/ksm/books/Nadiynist.pdf> – Назва з екрану.
69. Комп'ютерні мережі загальний принцип функціонування комп'ютерних мереж [Електронний ресурс]. – Режим доступу: http://vuzlib.com.ua/articles/book/21853-Komp%27juterni_merezhi_zagalnÑ/20.html – Назва з екрану.
70. Студопедия [Електронний ресурс]. – Режим доступу: http://studopedia.su/17_16565_problemi-bezpeki-suchasnih-korporativnih-merezh.html – Назва з екрану.
71. Сучасний захист інформації [Електронний ресурс]. – Режим доступу: <http://journals.dut.edu.ua/index.php/dataprotect> – Назва з екрану.

ДОДАТКИ

Додаток А
(обов'язковий)
Технічне завдання

ЗАТВЕРДЖЕНО
Зав. кафедри АІТ
_____ Кветний Р. Н.
«__» _____ 2019 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
« Розробка програмних засобів для системи відстеження рухомих об'єктів у відео »
08-02.МКР.000.00.001 ТЗ

Керівник роботи:
к. т. н., доц. каф. АІТ. Маслій Р. В.
“__” _____ 2019 р.

Виконавець: ст. гр. 1АКІТ-18м
Бондарчук А. С.
“__” _____ 2019 р.

Вінниця 2019

1. Назва та галузь застосування

Магістерська кваліфікаційна робота: «Розробка програмних засобів для системи відстеження рухомих об'єктів у відео». Галузь застосування – системи відстеження об'єктів.

2. Підстава для розробки є програмні засоби для підвищення ефективності систем відеоспостереження та відстеження об'єктів у реальному часі.

У системах відстеження об'єктів, не зважаючи на дуже велику кількість корисних функцій, є ряд недоліків, головними з яких є велика частота кадрів, а також велика затрата пам'яті для збереження відеоматеріалу. Підставою для розробки є наказ ВНТУ №254 від 02.10.2019.

3. Мета та призначення розробки

Зменшення розміру пам'яті, затраченої на збереження відеоматеріалів, а також розробка програмних засобів розпізнавання руху об'єкта для інтеграції у системи відеоспостереження.

4. Джерела розробки

1. Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016.

2. Системи відеоспостереження [Електронний ресурс]. – Режим доступу: <http://www.bastion.kiev.ua/Main/Videonablyudenie> – Назва з екрану.

3. MOTChallenge leaderboard, 2016. URL <https://motchallenge.net/results/MOT16/>. Accessed: May 24, 2017.

4. Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple on- line and realtime tracking. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3464–3468. IEEE, 2016.

5. НТКП ВНТУ. Факультет комп'ютерних систем і автоматики XLVII Науково-технічна конференція факультету комп'ютерних систем і автоматики. – [Електронний ресурс]. – Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2018/paper/view/4545/4613>. – Назва з екрану.

5. Показники призначення

Запропоновано новий метод відстеження об'єктів у відео, який на відміну від існуючих використовує нейромережеву модель уоло для відстеження об'єктів та фільтр Калмана для передбачення місцезнаходження об'єктів у нових кадрах що дозволяє підвищити вірогідність відстеження при короткочасних оклюзіях.

6. Економічні показники

- прогнозовані витрати на розробку – не більше 60 тис. грн;
- абсолютна ефективність розробки – не менше 200 тис. грн;
- термін окупності витрат для виробника – не більше 1,5 року.

7. Стадії розробки

Розділ 1 «Аналіз систем відеоспостереження» має бути виконаний до 03.10.2019.

Розділ 2 «Дослідження методів відстеження об'єктів у реальному часі» має бути виконаний до 25.10.2019.

Розділ 3 «Розробка та тестування програмних засобів для реалізації системи відстеження об'єктів» має бути виконаний до 11.11.2019.

Розділ 4 «Результати дослідження» мають бути виконані до 15.11.2019

Економічний розділ має бути виконаний до 3.12.2019.

8. Порядок контролю та приймання

Попередній захист магістерської кваліфікаційної роботи провести до 4.12.2019.

Додаток Б (обов'язковий).

Графічна частина

Завідувач каф. АІТ

д.т.н., професор

(підпис)

Кветний Р.Н

Науковий керівник

к.т.н., доцент

(підпис)

Маслій Р. В.

Тех. Контроль

к.т.н., доцент

(підпис)

Маслій Р. В.

Нормконтроль

к.т.н., доцент

(підпис)

Маслій Р. В.

Рецензент

(підпис)

(науковий ступінь, вчене звання, ініціали та
прізвище)

Студент гр. 1АКІТ-18м

Бондарчук А.С.

(підпис)

(ініціали та прізвище)

Продовження додатку Б

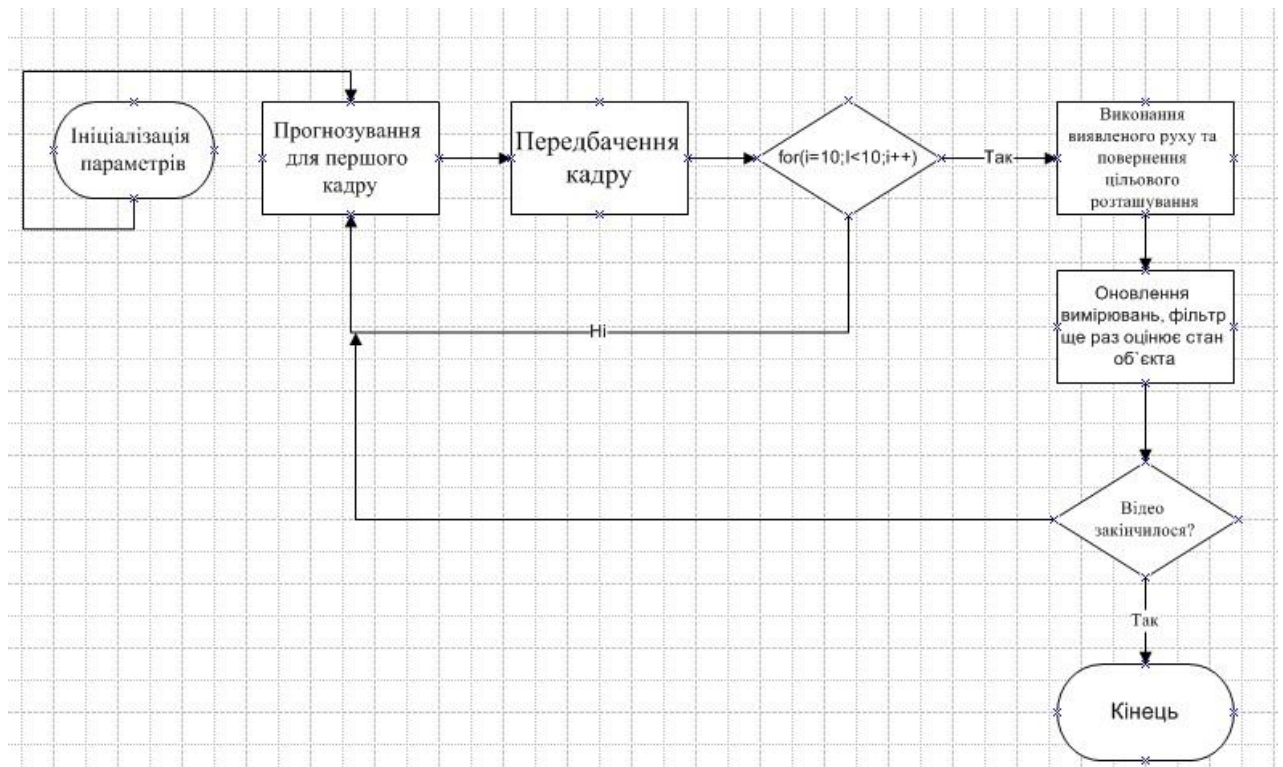


Рисунок Б.1 – Схема роботи фільтру Калмана при частоті 10 кадрів в секунду

Продовження додатку Б



Рисунок Б.2 – Структурна схема роботи фільтру Калмана

Продовження додатку Б

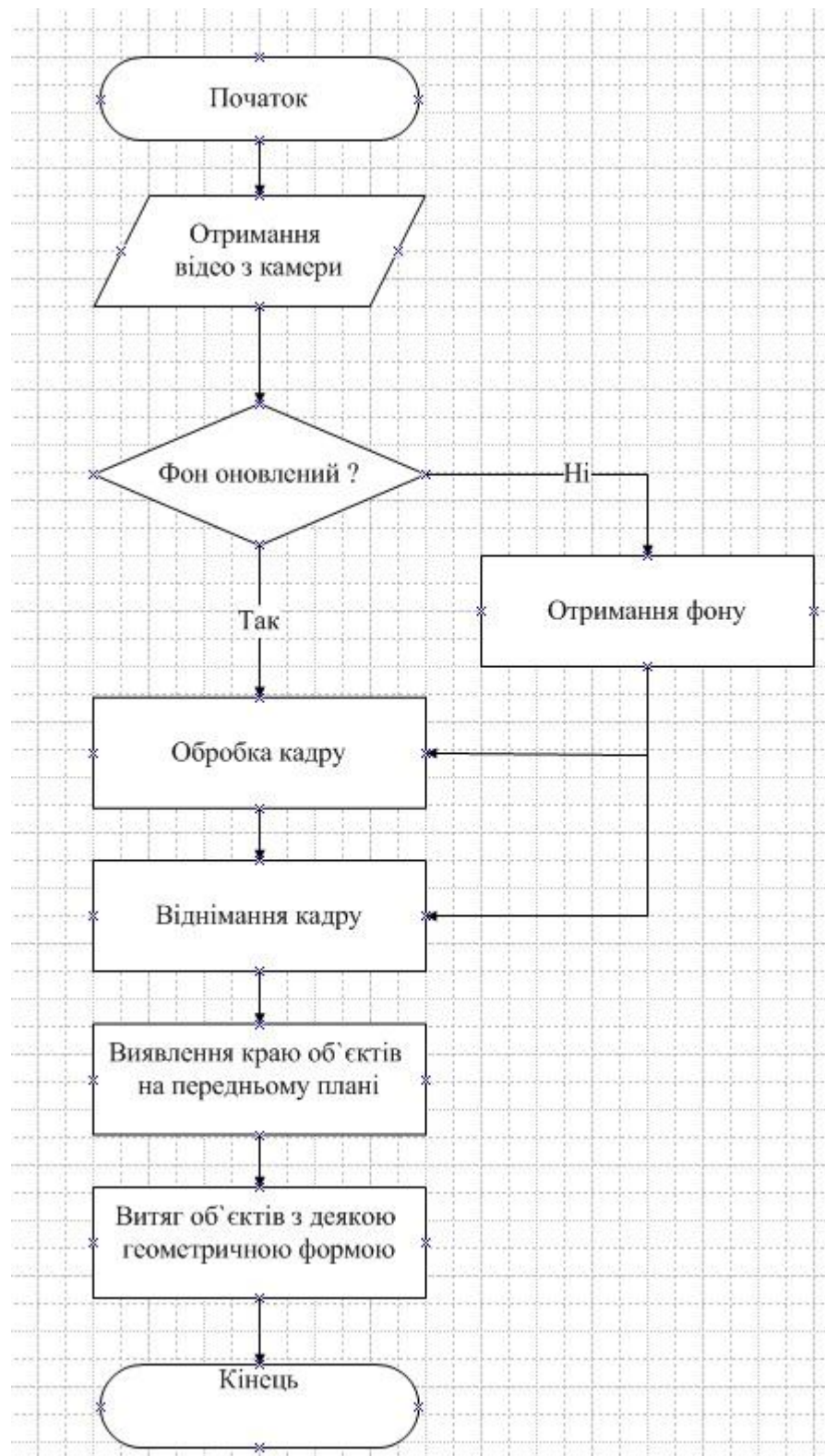


Рисунок Б.3 – Алгоритм роботи сегментації зображення

Продовження додатку Б

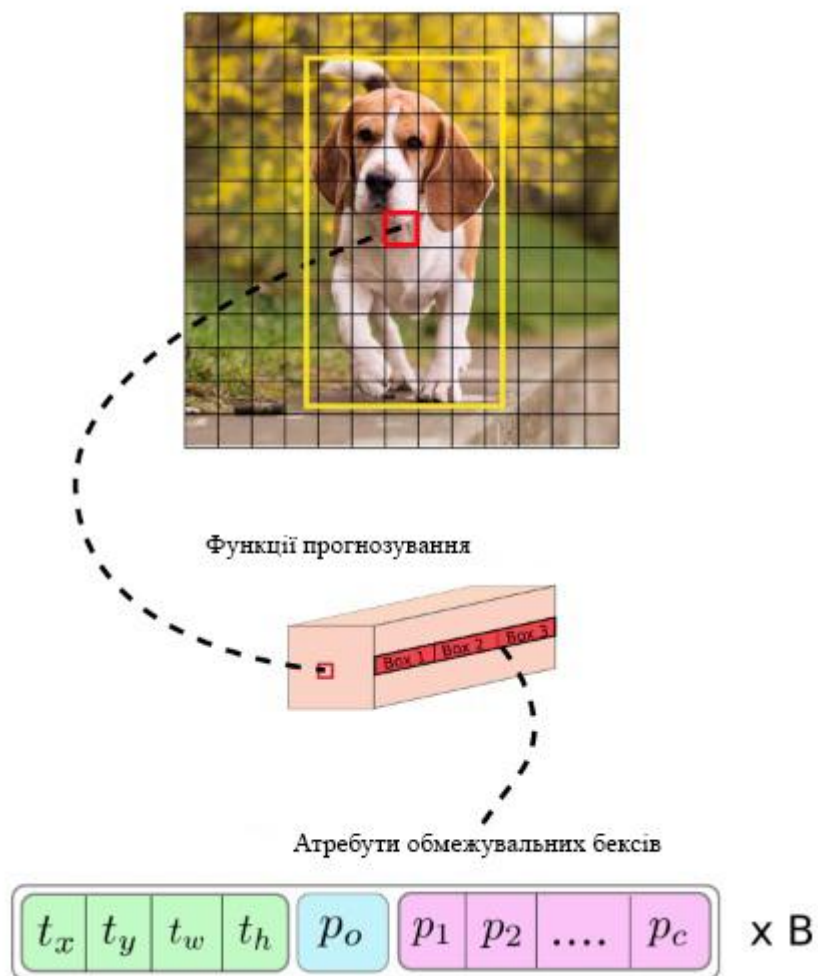


Рисунок Б.4 – Загальна структура роботи неймережевої моделі YOLO. Червоний квадрат відповідає за виявлення тварини за допомогою неймережевої моделі YOLO.

Продовження додатку Б

Intv	Rcll	Prcn	FAR	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP
1	29.5	100.0	0.00	226	1	163	62	0	51736	8156	5960	18.3	99.5
2	36.0	100.0	0.00	226	1	176	49	0	23460	3758	3134	25.8	99.6
3	39.7	100.0	0.00	226	1	183	42	0	14749	2520	2193	29.4	99.6
5	40.9	100.0	0.00	226	3	178	45	0	8679	1433	1290	31.1	99.7
10	33.6	100.0	0.00	226	0	119	107	0	4881	636	596	24.9	99.7
15	33.7	100.0	0.00	222	1	98	123	0	3242	391	336	25.8	99.8
30	22.3	100.0	0.00	221	0	64	157	0	1910	157	116	15.9	99.7

Рисунок Б.5 – Результати використання основних даних зі стаціонарними передбачувачами та лінійною вартістю, яку дає MOTChallenge DevKit.

Intv	Rcll	Prcn	FAR	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP
1	98.3	100.0	0.00	226	207	9	10	1	1245	46	38	98.2	95.1
2	91.7	100.0	0.00	226	154	38	34	12	3041	89	81	91.4	91.6
3	84.3	99.9	0.00	226	122	43	61	11	3844	151	135	83.6	89.0
5	68.9	99.8	0.01	226	72	51	103	17	4561	277	244	66.9	85.8
10	37.3	99.9	0.00	226	10	68	148	2	4605	334	298	32.7	83.6
15	21.9	99.9	0.00	222	1	43	178	1	3820	215	180	17.5	83.8
30	9.5	100.0	0.00	221	0	29	192	0	2225	67	34	6.8	88.9

Рисунок Б.6 – Результати використання основної істини за допомогою фільтрів Калмана та вартості IoU, яку дає MOTChallenge DevKit.

Додаток В (обов'язковий)

Лістинг програми

Algorithm 1 C++SORT

```

Set values for threshold, maxAge and minHitStreak
for all frames do
  Get detections from frame
  Remove detections with confidence less than threshold
  Predict new locations of objects using predictors
  Associate predictors and detections based on similarity
  for all predictors do
    if predictor has associated detection then
      PROPAGATE(predictor, detection)
    else
      PROPAGATE(predictor)
    end if
  end for
  for all unmatchedDetections do
    Initialise new predictor with detection
  end for
  Remove predictors that have had no associated detection the last maxAge frames
  for all predictors do
    if predictor had associated detection this and the last minHitStreak frames then
      Add state of predictor to trackings
    end if
  end for
  return trackings
end for

```

```
#Initialize deep sort object.
```

```
deepsort = deepsort_rbc(wt_path='ckpts/model640.pt') #path to the feature extractor
model.
```

```
#Obtain all the detections for the given frame.
```

```
detections,out_scores = get_gt(frame,frame_id,gt_dict)
```

```
#Pass detections to the deepsort object and obtain the track information.
```

```
tracker,detections_class = deepsort.run_deep_sort(frame,out_scores,detections)
```



```
#Obtain info from the tracks.
```

```
for track in tracker.tracks:
```

```
    bbox = track.to_tlbr() #Get the corrected/predicted bounding box
```

```
    id_num = str(track.track_id) #Get the ID for the particular track.
```

```
    features = track.features #Get the feature vector corresponding to the detection.
```

```
class deepsort_rbc():
```

```
    def __init__(self):
```

```
        #loading this encoder is slow, should be done only once.
```

```
        self.encoder = torch.load('model640.pt')
```

```
        self.encoder = self.encoder.cuda()
```

```
        self.encoder = self.encoder.eval()
```

```
        print("Deep sort model loaded")
```

```
        self.metric = nn_matching.NearestNeighborDistanceMetric("cosine",.5,
```

```
100)
```

```
        self.tracker= Tracker(self.metric)
```

```
        self.gaussian_mask = get_gaussian_mask().cuda()
```

