Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем управління
(повна назва кафедри)

Бакалаврська дипломна робота
на тему:

«АВТОМАТИЗОВАНА СИСТЕМА ОПРАЦЮВАННЯ СЕНСОРНОЇ ВІДЕОІНФОРМАЦІЇ ДЛЯ КОНТРОЛЮ ВІДСТАНІ МІЖ ОБ'ЄКТАМИ»»

Виконав: студент 4-го курсу,
групи АКІТ-22мс
спеціальності 151 –
Автоматизація такомп'ютерно-інтегровані технології
(шифр і назва спеціальності)

Дмитро ШТУРМА
(ім'я та прізвище)

Керівник: к.т.н., доцент каф. КСУ
Олена Никитенко
(ім'я та прізвище)
«10» 06 2024р.

Рецензент: к.т.н., доц. каф. АІІТ
Юрій ІВАНОВ
(ім'я та прізвище)
«11» червня 2024 р.

Допущено до захисту
Завідувач кафедри КСУ
д.т.н., проф.
В'ячеслав КОВТУН
(ім'я та прізвище)
«14» червня 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет Інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти І (бакалаврський)
Галузь знань – 15 Автоматизація та приладобудування
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
Освітньо-професійна програма Інтелектуальні комп'ютерні системи управління

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
д.т.н., проф.
_____В'ячеслав КОВТУН
«24» травня 2024 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Штурмі Дмитру Ярославовичу
(прізвище, ім'я, по батькові)

1. Тема роботи – «Автоматизована система опрацювання сенсорної відеоінформації для контролю відстані між об'єктами»
Керівник роботи: Никитенко Олена Дмитрівна, к.т.н., доцент, затверджені наказом вищого навчального закладу від 11 березня 2024 р. № 80
2. Строк подання студентом роботи 10 червня 2024 р.
3.Вихідні дані до роботи: Інтегроване середовище розробки (IDE): IntelliJ IDEA, мова програмування: Java, Фреймворки:
Sharma, A. (2022). "Spring Boot: Up and Running." Електронний ресурс (доступ: 2 червня 2024).
Long, J. (2021). "Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools." Електронний ресурс (доступ: 3 червня 2024).
Spring Framework для побудови backend частини.
Spring Boot для швидкого розгортання та конфігурації додатку.
4. Зміст розрахунково-пояснювальної записки: вступ; аналіз та постановка задачі; розробка архітектури та алгоритмів програмного продукту; розробка програмного продукту; тестування програми, висновки; перелік посилань, додатки.
5. Перелік графічного матеріалу: мета; об'єкт та предмет дослідження; діаграма варіантів використання; структурна схема компонентів додатку; загальний алгоритм роботи;.

6 Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Виконанняприйняв
1-3	д.т.н., професор каф. КСУ Ковтун В.В.		

Дата видачі завдання 29.05.2024 року

7 КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі дослідження	10.05.2024 21.05.2024	
2	Розробка програмного забезпечення системи	22.05.2024 - 27.05.2024	
3	Тестування програмного забезпечення	28.05.2024 - 30.05.2024	
4	Графічні матеріали	01.06.2024 - 03.06.2024	
5	Оформлення пояснювальної записки та презентації	04.06.2024 - 09.06.2024	

Студент _____ Дмитро ШТУРМА
(підпис)

Керівник роботи _____ Олена НИКИТЕНКО
(підпис)

АНОТАЦІЯ

Цю дипломну роботу присвячений вирішенню проблеми вимірювання відстані до об'єктів на відео.

Метою цієї роботи є не лише опис рішень для відеовиміру відстані, але й дослідження їх ефективності та потенціалу застосування. У результаті аналізу та розробки було натреновано нейронну мережу, здатну визначати об’єкти на відео. Крім того, розроблено алгоритм для визначення відстані до об’єктів, що використовує дані про камеру, об’єкт та інформацію з нейромережі. Система має веб-інтерфейс для зручної взаємодії з користувачами. Розроблена система може використовуватися не лише для розпізнавання відстані до об’єктів на записаному відео, але й на відео в режимі реального часу.

Ключові слова: Python, OpenCV, PyTorch і Werkzeug, веб-інтерфейс, Flask.

‘

ANNOTATION

This thesis is devoted to solving the problem of measuring the distance to objects on video.

The purpose of this work is not only to describe solutions for video distance measurement, but also to study their effectiveness and application potential. As a result of the analysis and development, a neural network capable of identifying objects in the video was trained. In addition, an algorithm has been developed to determine the distance to objects, which uses data about the camera, the object, and information from a neural network. The system has a web interface for convenient interaction with users. The developed system can be used not only to recognize the distance to objects in recorded video, but also in real-time video.

Keywords: Python, OpenCV, PyTorch and Werkzeug, web interface, Flask.

‘

ЗМІСТ

ВСТУП

У сучасному світі технологій, що швидко розвиваються, обробка візуальних даних відіграє все більш важливу роль у різних сферах: від безпілотних автомобілів до систем безпеки та спостереження. У цьому випадку одним із важливих завдань є визначення відстані до об’єктів на відео, що є необхідним етапом аналізу та прийняття рішення.

Дана робота присвячена дослідженню та розробці системи вимірювання відстані до об'єктів на відео. Метою цієї роботи є створення ефективного та точного рішення, яке можна використовувати в різних сферах, де вимірювання відстані є важливим елементом, наприклад, транспортні засоби з можливостями автономного водіння, системи контролю та безпеки, розумні міста (подібні до інтелектуальних систем керування) в Чикаго) та багато інших.

Для досягнення цієї мети цей документ складається з трьох основних частин. Глава 1 містить огляд концепцій, методів і систем, пов’язаних із відеодальністю.

Розділ 2 детально розглядає методи, алгоритми та прийоми, що використовуються для розробки систем вимірювання відстані.

Розділ 3 представляє процес розробки системи, включаючи обробку та вибір даних, навчання моделі, оцінку продуктивності та розробку веб-інтерфейсу для взаємодії з моделлю.

Метою цієї роботи є не лише опис рішень для відеовиміру відстані, але й дослідження їх ефективності та потенціалу застосування. Очікується, що результати цієї роботи не тільки розширять загальні знання про вимірювання відстані до об’єктів у відео, але й сприятимуть подальшому розвитку систем аналізу візуальних даних і забезпечать безпеку та ефективність у різних сферах.

Задачі дослідження:

- зробити огляд аналогів об’єкта дослідження,
- проаналізувати аналоги й визначити параметри та характеристики створюваної системи,
- спроектувати систему, орієнтовану на досягнення поставленої мети,
- протестувати створену систему та довести її ефективність в порівняння зі знайденими аналогами.

Об’єкт дослідження – процес опрацювання сенсорної відеоінформації для контролю відстані між об’єктами.

Предмет дослідження – методи й засоби штучного інтелекту та комп’ютерного зору.

Інноваційність: Автоматизована система опрацювання сенсорної відеоінформації для контролю відстані між об'єктами використовує передові технології комп'ютерного зору та штучного інтелекту, що дозволяє аналізувати відеоінформацію в режимі реального часу з високою точністю. Інтеграція різних типів сенсорів, таких як LiDAR, ультразвукові сенсори та стереокамери, забезпечує багатошарове збирання даних, що значно підвищує надійність і точність вимірювань.

Практична цінність системи полягає у її широкому спектрі застосувань, включаючи безпеку на транспорті, автоматизацію промислових процесів та управління натовпами. Наприклад, у транспортній сфері система допомагає запобігати зіткненням, автоматично контролюючи відстань між транспортними засобами. У промисловості вона може використовуватись для моніторингу та управління робототехнічними системами, забезпечуючи безпеку та ефективність виробничих процесів.

1 ДОСЛІДЖЕННЯ Й АНАЛІЗ АНАЛОГІВ ДЛЯ ВИМІРЮВАННЯ ВІДСТАНІ НА ВІДЕО

1.1 Визначення поняття відстані та її вимірювання на відео

1.1.1 Поняття дистанції та її значення в різних сферах

Як визначено в джерелі [1], відстань — це міра відстані між об’єктами або точками. У фізиці чи повсякденному використанні відстань може являти собою фізичну довжину або оцінку на основі інших критеріїв. Існують різні способи визначення відстані між фізичними об’єктами в різних ситуаціях. Пряма або евклідова відстань — це відстань між двома точками фізичного простору і дорівнює довжині прямої між ними, тобто найкоротшого шляху. Відстань по прямій математично формалізується як евклідова відстань у двох і трьох вимірах. У евклідовій геометрії відстань між двома точками A і B зазвичай позначається як |AB|. У координатній геометрії евклідова відстань обчислюється за допомогою теореми Піфагора. Відстань між точками (x1, y1) і (x2, y2) на площині визначається як:

$$d = \sqrt{((\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2)} = \\ = \sqrt{((x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2)}$$

1.1.2 Методи відеолокації та їх характеристика

Існує багато способів вимірювання відстані по прямій лінії, наприклад за допомогою лінійки або опосередковано за допомогою радара (для великих відстаней) або інтерферометрії (для дуже коротких відстаней). Джерело [1] згадує «Космічні сходи», набір методів для вимірювання надзвичайно великих відстаней.

Деякі способи вимірювання відстані відео включають:

• Піксельний метод:

о Примітка. Відстань визначається кількістю пікселів, зайнятих об’єктом у кадрі відео.

о Особливості: метод простий, але він схильний до помилок через зміни кута зйомки, розміру об’єкта та відстані.

• Масштабно-інваріантне перетворення ознак (SIFT):

о Опис: порівняйте об’єкти на зображенні з базою даних відомих об’єктів, щоб оцінити відстань.

о Особливості: точніші, ніж піксельні методи, але потребують великих обчислювальних ресурсів і великої бази даних відомих об’єктів.

• Структура з руху (SFM):

о Опис: використовуйте серію зображень, щоб створити 3D-модель середовища та оцінити відстані.

о Характеристики: метод є точним, але вимагає великої кількості даних і обчислювальних ресурсів.

• світловий потік:

о Опис: вимірювання руху пікселів у послідовних відеокадрах для оцінки відстані.

о Характеристики: швидкий і відносно точний, але схильний до помилок через зміни тіней і освітлення об’єктів.

• Методи на основі машинного навчання (наприклад, YOLOvN):

о Опис: використовуйте модель глибокого навчання для виявлення об’єктів і оцінки відстані на основі розміру та положення об’єкта у відеокадрі.

о Особливості: Висока точність, але вимагає великої кількості навчальних наборів даних і великої кількості обчислювальних ресурсів.

‘

1.2 Огляд існуючих систем відеоспостереження

1.2.1 Огляд систем, що використовуються в різних галузях промисловості

Відповідно до посилання в джерелі [2], існує кілька існуючих систем вимірювання відстані.

Один із них — це алгоритм карти стереоскопічної диспаратності для оцінки відстані між двома камерами та об’єктом.

Іншим підходом є автономна система вимірювання відстані транспортного засобу за допомогою стереокамер, яка використовує зображення зі стереокамери для оцінки відстані між камерою та об’єктом.

Іншим методом є монокулярна оцінка відстані за допомогою апроксимації точкової камери, щоб уникнути нещасних випадків і аварій з перекиданням, яка використовує апроксимацію точкової камери для оцінки відстані між камерою та об’єктом, щоб уникнути аварійних ситуацій.

Наступний метод — багатопроменева оптична решітка для далекобійної передачі лідарних і космічних даних з використанням багатопроменевої оптичної фазованої решітки для передачі лідарних і космічних даних.

Відповідно до літератури [3], фокусне відеовимірювання відстані є методом визначення відстані в полі одного огляду. Цей спосіб передбачає фіксацію об'єктива в положенні, що відповідає максимальній різкості Отримане зображення називається глибиною різкості (DFF). DFF — це відстань між найближчим і найдальшим об’єктом сцени, яка забезпечує прийнятну чіткість зображення. За допомогою глибини різкості можна оцінити відстань до об’єкта від камери.

Інше джерело [4] обговорює методи вимірювання відстані рухомих об’єктів у відео. У цій статті використовується метод фонової різниці для виділення цільового об’єкта, а потім обчислюється відстань між цільовим

об’єктом і камерою на основі розміру цільового об’єкта. Цей метод можна використовувати для вимірювання відстані рухомих об’єктів у відео.

Існують різноманітні системи вимірювання відстані, у тому числі ті, що базуються на стереоскопічних картах невідповідності зображень, апроксимації точкових отворів, багатопроменевих оптичних фазованих ґратках і трьох алгоритмах положення фокусу. Кожен із цих методів має переваги та недоліки, і вибір методу залежить від конкретних вимог застосування.

1.2.2 Характеристика існуючих систем, їх переваги та недоліки

Характеристики, переваги та недоліки існуючих систем відеоспостереження залежать від конкретних методів, що використовуються. Наприклад, система, запропонована в [5], виявляє транспортні засоби за допомогою двох камер, встановлених як одна стереокамера, і обчислює відстань між транспортними засобами за допомогою геометричних похідних і додаткових технічних даних. Перевагами цієї системи є висока точність і ефективне вимірювання в реальному часі, але недоліком є те, що вона вимагає спеціального обладнання та технічних знань.

У статті [6] пропонується система для людей з вадами зору, яка використовує виявлення об’єктів для оцінки відстані. Перевагами цієї системи є точність і короткий час розпізнавання, але недоліком є те, що вона може не підходити для всіх застосувань.

Система відеовимірювання відстані, запропонована в [7], використовує систему відеовимірювання відстані (VDM), яка є такою ж точною, як і традиційна система EDM. Перевагою цієї системи є те, що вона дозволяє точніше ідентифікувати точки вимірювання та не вимагає використання інструментів (цвяхів), які можуть зміщувати пісок навколо позначок, недоліком є те, що вона може не підходити для всіх застосувань.

Загалом, характеристики, переваги та недоліки існуючих систем відеоспостереження залежать від конкретного використовуваного методу та вимог застосування. Кожна система має свої переваги та недоліки, і вибір системи залежить від конкретних потреб проекту.

1.3 Методи ранжування та аналіз алгоритмів відео

1.3.1 Приклади методів і алгоритмів відеолокації

У статті [8] метод подібності трикутника використовується для розрахунку відстані від об’єкта на відео до камери. Метод передбачає знання фокусної відстані камери, висоти об’єкта та радіуса круглої позначки на об’єкті. Відстань можна розрахувати за такою формулою:

$$d = \frac{h * f}{r}$$

де d — відстань, h — висота об’єкта, f — фокусна відстань камери, r — радіус круглого маркера.

Одним із інших методів визначення відстані до відео є метод комплексного логарифмічного відображення (CLM), який передбачає перетворення зображення з ортогональної системи координат у полярну систему координат і визначення співвідношення між двома фотографіями на основі властивостей концентричних кіл. . Інший підхід передбачає оцінку глибини за допомогою монокулярної камери

Параметри камери та геометрія зображення. Третій спосіб полягає у використанні камери зі змінним кутом нахилу для вимірювання відстані до об’єкта шляхом оптимізації квадратури.

Згідно з джерелом [8], вимірювання відстані важливе для різних застосувань, наприклад, для надання інформації про виявлені об’єкти та їх абсолютну відстань у сцені, зафіксованій системою перед автомобілем, або для автономних транспортних засобів, багатопілотних транспортних

засобів. машин і роботів. Інформація про відстань також необхідна комп'ютерам для розпізнавання навколишнього середовища та є важливою технологією для ідентифікації та вимірювання відстані до інших об'єктів поблизу.

Інші технології, які зараз використовуються для визначення відстані, включають радар і лідар. Радар використовує електромагнітні хвилі для обчислення відстані, а лідар використовує відбиття променя для вимірювання відстані, форми та матеріалу середовища.

Загалом, існують різні методи й алгоритми для визначення відстані на відео залежно від конкретного застосування, наявного обладнання та технічних знань. Подібні трикутники, CLM, оцінка монокулярної камери та методи камери зі змінним кутом нахилу – лише деякі приклади багатьох доступних методів.

1.3.2 Характеристика методів і алгоритмів, їх переваги і недоліки

Характеристики методів і алгоритмів, які використовуються для визначення відстаней у відеозаписах, залежать від конкретного використовуваного методу. Наприклад, традиційні методи оцінки відстані вимагають комплексного калібрування внутрішніх і зовнішніх параметрів камери, як описано в статті [4]. Нові методи, засновані на нейронних мережах, можуть підвищити швидкість і точність вимірювання відстані, але можуть вимагати значних обчислювальних ресурсів і технічного досвіду.

У цьому розділі вивчаються та аналізуються різні аспекти вимірювання відстані до об'єктів у відео, а також оглядаються існуючі системи та методи, які використовуються для цієї мети.

У ході роботи було визначено поняття відстані та її значення в різних сферах. Також розглядаються різні методи відеолокації та їх характеристики.

Далі надається огляд існуючих систем відеодальності, особливо в

різних сферах застосування. Розглянуто характеристики цих систем, їх переваги та недоліки.

Також аналізуються методи та алгоритми. Розглянуто приклади використання, а також опис переваг і недоліків різних підходів.

Загалом у цьому розділі розглядається концепція відстані, методи вимірювання відстані за допомогою відео, а також розглядаються існуючі системи та методи визначення відстані. Цей розділ містить необхідну базу знань для подальшого розвитку систем вимірювання відстаней до об'єктів у відео.

2 МЕТОДИКИ, ТЕХНОЛОГІЇ ТА ПІДХОДИ ДЛЯ СТВОРЕННЯ СИСТЕМИ ВИМІРЮВАННЯ ВІДСТАНІ НА ВІДЕО

2.1 Огляд бібліотеки OpenCV та її функцій

2.1.1 Загальна характеристика OpenCV та його застосування в комп'ютерному зорі

OpenCV — це широко використовувана бібліотека комп’ютерного бачення з відкритим кодом, розроблена компанією Intel, яка забезпечує повний набір можливостей обробки зображень для комп’ютерного бачення в режимі реального часу. Він доступний безкоштовно за ліцензією Apache 2 і працює на найпопулярніших операційних системах, таких як GNU/Linux, OS X, Windows, Android та iOS, що робить його дуже універсальним інструментом для розробників. Інструмент OpenCV особливо корисний для машинного навчання комп’ютерного зору та містить повну бібліотеку машинного навчання загального призначення, зосереджену на статистичному розпізнаванні образів і кластеризації. Бібліотека оптимізована для задач машинного зору, що потребують інтенсивних обчислень, і може використовувати переваги багатоядерних процесорів. Він також містить понад 2500 алгоритмів, обширну документацію та приклади коду для комп’ютерного зору в реальному часі. [9] OpenCV написано мовами C і C++ і зараз активно розробляє інтерфейси для Python, Ruby, Matlab та інших мов. Він також пропонує апаратне прискорення NVIDIA CUDA і графічних процесорів (GPU) і Open Compute Language (OpenCL), що робить його ефективним і ефективним інструментом для операцій у реальному часі. Бібліотека OpenCV є популярним інструментом, який використовують великі підприємства та державні установи, такі як Google, Toyota, IBM, Microsoft, Sony, Siemens і Facebook. Відомі стартапи комп’ютерного бачення використовують його для створення потужних

продуктів комп’ютерного бачення та рішень штучного інтелекту. OpenCV використовується багатьма дослідницькими центрами, включаючи Стенфордський університет, MIT, INRIA, Кембриджський університет і Університет Карнегі-Меллона.

Бібліотека OpenCV зосереджена на додатках штучного зору в реальному часі та надає прості у використанні інструменти для цього завдання, надаючи понад 500 функцій, що охоплюють різні сфери, щоб допомогти людям швидко створювати складні програми. OpenCV також активно використовується для контролю продукту на виробництві, медичної візуалізації, аналізу безпеки, людино-машинного інтерфейсу, калібрування камери, стереобачення (3D-бачення) і роботизованого бачення. [9]

OpenCV є найбільш універсальним інструментом розробки комп’ютерного зору. Бібліотека використовується для всього, від розпізнавання зображень, 2D або 3D аналізу, відстеження руху до розпізнавання облич. Крім того, бібліотека активно використовується в техніках виявлення об’єктів для розпізнавання зображень і визначення місцезнаходження певних об’єктів у відеоданих або зображеннях, таких як автомобілі, люди, тварини, а також окремі частини чи обладнання в промисловому виробництві. Бібліотека OpenCV також використовується для сегментації зображень, де застосовуються алгоритми обробки зображень, щоб розділити зображення на різні частини. Сегментація часто використовується для спрощення, заміни або покращення зображень, часто в поєднанні з іншими завданнями комп’ютерного зору. OpenCV також використовується для розпізнавання поз і жестів людини з метою їх інтерпретації та розуміння за допомогою аналізу відео. Інші можливості бібліотеки включають автоматичне розпізнавання обличчя, доповнену реальність і обчислення пози об’єктів, щоб надати спосіб зрозуміти, як об’єкти розташовані в 3D-просторі (наприклад, як вони обертаються). [9]

OpenCV використовується в багатьох програмах, продуктах і дослідницьких проектах. Ці програми включають зшивання зображень із камер на супутникових або мережевих картах, вирівнювання відсканованих зображень, зменшення шуму медичних зображень, аналіз об’єктів, системи безпеки, спостереження та виявлення вторгнень, автоматизовані системи моніторингу, системи керування виробництвом штучного інтелекту, калібрування камер, оборонні та військові застосування, а також дрони, наземні та підводні апарати. Інструмент навіть використовувався для розпізнавання звуку та музики, де методи візуального розпізнавання застосовувалися до зображень звукових спектрограм. [9]

Останнім часом розробка без коду або з низьким кодом стала новим способом для компаній і організацій швидше та ефективніше надавати та підтримувати рішення. Розвиток комп’ютерного зору часто дуже складний і вимагає кількох ітерацій розробки. Таким чином, реалізації комп’ютерного бачення значно виграють від можливості візуального проектування та розгортання технології без коду в автоматизований спосіб. Платформа комп’ютерного бачення Viso Suite надає функціональні можливості OpenCV як модульні будівельні блоки, які можна використовувати для швидкого створення програм комп’ютерного бачення без кодування з нуля. Це дозволяє командам розробників швидше використовувати бібліотеку та полегшує інтеграцію з різним обладнанням, таким як камери, периферійні пристрої та моделі машинного навчання.

Бібліотеки OpenCV можна створювати для незліченних випадків використання, включаючи аналіз медичних зображень для підтвердження діагнозу людини, розпізнавання телевізійних реклам або логотипів за допомогою штучного зору, відстеження спортсменів у спорті та фітнесі, розпізнавання сцен і аналіз продуктивності, а також підрахунок людей, які займаються спортом у громадських місцях, наприклад як аеропорти, роботизована автоматизація з інтелектуальними інтерфейсами на основі

зору, автоматичний перегляд і аналіз відео з використанням безперервного комп’ютерного бачення, пошук зображень на цифрових платформах, виявлення дефектів або несправностей у виробничих процесах, підрахунок транспортних засобів на автомагістралях і програми з камерами відеоспостереження. що виявляє фізичне насильство, напади та порушення ПДР.

Отже, OpenCV — це загальна бібліотека комп’ютерного зору з відкритим кодом, яка надає повний набір можливостей обробки зображень для комп’ютерного бачення в реальному часі. Його широко використовують великі підприємства, державні установи, стартапи та дослідницькі центри. Бібліотека написана на C і C++ і працює на найпопулярніших операційних системах, таких як GNU/Linux, OS X, Windows, Android та iOS. Крім того, бібліотека дуже орієнтована на роботу в режимі реального часу.

2.1.2 Огляд основних функцій і алгоритмів OpenCV для обробки відео

OpenCV містить модуль аналізу відео, який включає алгоритми оцінки руху, віднімання фону та відстеження об’єктів. Алгоритми оцінки руху використовуються для визначення руху об’єктів у відеорядах. Він реалізований за допомогою методу блокового відображення, який порівнює пікселі в двох послідовних відеокадрах. Алгоритми віднімання фону використовуються для відділення об’єктів на передньому плані від фону. Він використовується для виявлення рухомих об’єктів на відео та реалізований за допомогою моделі суміші Гауса (GMM). Алгоритми відстеження об’єктів використовуються для відстеження об’єктів у відеорядах. Він реалізований за допомогою фільтра Калмана, який передбачає розташування об'єкта в наступному кадрі відео. [10]

OpenCV також надає модуль відеовведення/виведення (videoio), який містить простий у використанні інтерфейс для захоплення відео та

відеокодеків. Модуль підтримує читання і запис відео в декількох форматах, включаючи AVI, MPEG і MP4. Він також підтримує захоплення відео в реальному часі з камер і відеопотоків. Цей модуль надає функціональність для налаштування властивостей захоплення відео (таких як роздільна здатність, частота кадрів і кодек) і для отримання інформації про відео (такої як кількість кадрів, частота кадрів і роздільна здатність). [10]

Щоб обробити відео за допомогою OpenCV у Python, ви створюєте об’єкт cv2.VideoCapture, який є класом для захоплення відео з відеофайлу, послідовності зображень або камери. Щоб відтворити відео, ви можете створити цикл while і зчитувати кожен кадр відео за допомогою функції cap.read(). Ця функція повертає логічне значення, яке вказує, чи прочитано кадр і сам кадр було успішним. Цикл продовжується, доки не залишиться кадрів для читання. У циклі можна виконувати різні завдання обробки відео, наприклад застосовувати фільтри, виявлення об’єктів або відстеження руху об’єктів.

2.2 Огляд архітектури та особливостей YOLOv8

2.2.1 Загальна характеристика мережевої архітектури YOLOv8

YOLOv8 — це остання модель виявлення об’єктів, яка базується на своїй попередниці з використанням нових методів і інструментів оптимізації. Остання версія моделі вийшла 10 січня 2024 року. Однією з ключових особливостей моделі є її настроювана архітектура, яка дозволяє користувачам легко змінювати структуру та параметри моделі відповідно до своїх потреб. Архітектура YOLOv8 складається з повністю згорткової нейронної мережі, яку можна розділити на дві основні частини: базову частину та верхню частину. [11]

Модель заснована на модифікованій версії архітектури CSPDarknet53,

яка складається з 53 згорткових рівнів і використовує техніку, що називається міжступеневими частковими з'єднаннями, щоб покращити потік інформації між різними рівнями мережі. YOLOv8 складається з кількох згорткових шарів зверху, за якими йде серія повністю пов'язаних шарів, відповідальних за прогнозування обмежувальних рамок, оцінок функцій і ймовірностей класів для функцій, виявлених на зображенні.

Одним із ключових удосконалень моделі YOLO є використання механізмів саморегулювання на вершині мережі. Цей механізм дозволяє моделі зосереджуватися на різних частинах зображення та регулювати важливість різних функцій на основі їх відповідності завданню. Іншою важливою особливістю є можливість виконувати багатомасштабне виявлення об'єктів, що досягається за допомогою мережі піраміди функцій. Мережа складається з кількох шарів, які виявляють об'єкти в різних масштабах, що дозволяє моделі виявляти великі та малі об'єкти на зображеннях.

YOLOv8 також підтримує різні магістралі, такі як EfficientNet, ResNet і CSPDarknet, що дає користувачам можливість вибрати найкращу модель для конкретного випадку використання. Крім того, модель використовує адаптивне навчання для оптимізації швидкості навчання та збалансування функції втрат, що призводить до кращої продуктивності моделі. Модель також використовує передові методи розширення даних, такі як MixUp і CutMix, для підвищення надійності та загальності.

Існує кілька моделей виявлення об'єктів, які вважаються передовими з точки зору точності, наприклад EfficientDet, DETR і Cascade R-CNN. Хоча YOLOv8 може конкурувати з цими моделями, вибір моделі для використання в кінцевому підсумку залежить від конкретного випадку використання та вимог.

Щоб використовувати YOLOv8, вам потрібен комп'ютер із графічним процесором, підтримка фреймворку глибокого навчання (наприклад,

PyTorch або TensorFlow) і доступ до YOLOv8 GitHub. Бібліотека програмного забезпечення YOLOv8 є відкритим кодом і доступна для досліджень і розробок на GitHub.

Як і його попередник, YOLOv8 розроблено для ефективної роботи на готовому апаратному забезпеченні, що робить його надійним рішенням для завдань виявлення об’єктів у реальному часі, зокрема в середовищах з обмеженими ресурсами. YOLOv8 досягає вищої точності, ніж попередні версії YOLO, і повністю конкурує з найсучаснішими моделями виявлення об’єктів. [12].

Рисунок 2.2 – Додаткові завдання, які може виконувати YOLOv8 [12]

Таким чином, YOLOv8 — це проста в налаштуванні модель виявлення об’єктів. Модель використовує модифіковану версію магістралі CSPDarknet53 із механізмом самоконтролю на вершині мережі для підвищення точності та виявлення об’єктів у різних масштабах. Модель підтримує різні базові мережі, адаптивне навчання та розширені методи розширення даних для покращення продуктивності моделі. Незважаючи на те, що YOLOv8 успішно конкурує з найсучаснішими моделями виявлення об’єктів, кінцева модель, яка буде обрана для використання, залежить від конкретного випадку використання та вимог.

2.2.2 Огляд основних функцій і переваг YOLOv8 для обробки відео

YOLOv8 — це найсучасніша модель виявлення об’єктів, яка

забезпечує неперевершену продуктивність з точки зору швидкості та точності [13]. Однією з головних переваг YOLOv8 для обробки відео є його підвищена швидкість, оскільки він забезпечує вищу швидкість обробки, ніж інші моделі виявлення об’єктів, зберігаючи високу точність. Це робить модель придатною для завдань обробки відео в реальному часі, таких як автоматичне керування автомобілями та системи спостереження. YOLOv8 також підтримує різні магістральні мережі, такі як

EfficientNet, ResNet і CSPDarknet, що дає користувачам можливість вибрати найкращу модель для конкретного випадку використання.

Ще однією перевагою YOLOv8 для обробки відео є його здатність виконувати багатомасштабне виявлення об’єктів, що досягається за допомогою пірамідної мережі функцій. Мережа складається з кількох шарів, які виявляють об’єкти в різних масштабах, що дозволяє моделі виявляти великі та малі об’єкти на зображеннях. Це особливо корисно для завдань обробки відео, де об’єкти можуть мати різні масштаби та розміри.

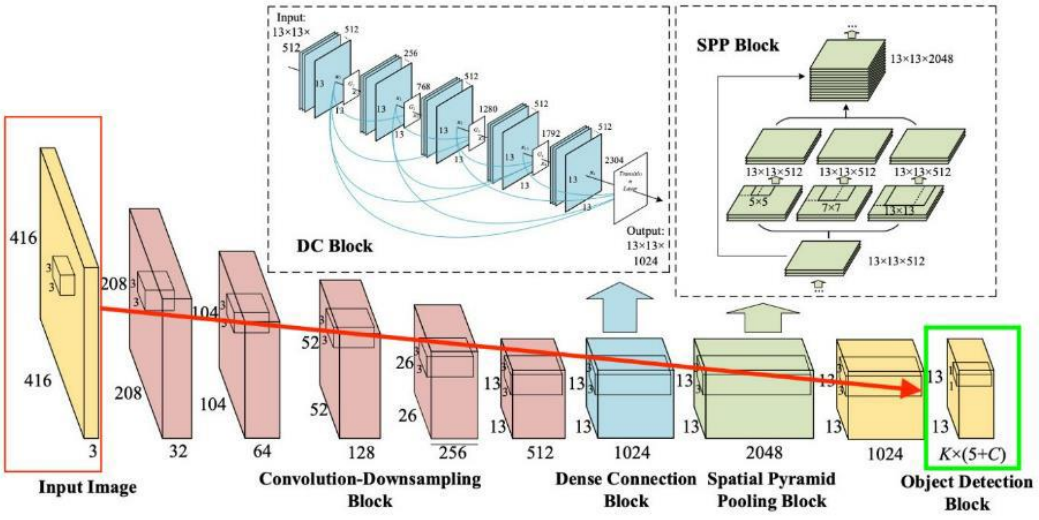


Рисунок 2.3 – Схема обробки зображення в YOLOv8[13]

Крім того, YOLOv8 використовує передові технології доповнення даних, такі як MixUp і CutMix, для підвищення надійності та узагальнення

моделі. Це важливо для завдань обробки відео, де модель повинна мати можливість виявляти об’єкти за різних умов освітлення, фону та орієнтації.

YOLOv8 також має широкі можливості налаштування, що дозволяє користувачам легко змінювати структуру та параметри моделі відповідно до своїх потреб. Це корисно для завдань обробки відео, вимоги до яких можуть відрізнятися залежно від конкретного випадку використання. YOLOv8 також надає попередньо підготовлені моделі для легкого використання та перенесення навчання в різні набори даних.

Загалом, YOLOv8 — це потужний і універсальний алгоритм виявлення об’єктів, який можна використовувати для завдань обробки відео в різних сценаріях реального світу, таких як автономні транспортні засоби, відеоспостереження та роботи. Більш висока швидкість, багатомасштабне виявлення об’єктів, передові методи розширення даних і настроювана архітектура роблять це рішення придатним для широкого спектру застосувань.

2.3 Огляд методів побудови наборів даних розпізнавання об’єктів у відео

2.3.1 Загальна характеристика методу побудови навчального набору даних мережі YOLOv8

Створення набору даних є важливим кроком у навчанні моделі YOLOv8 на наборі даних користувача [14]. Першим кроком у створенні набору даних є збір зображень і анотування їх обмежувальними рамками навколо об’єктів, які цікавлять користувача. Це можна зробити за допомогою різних інструментів анотації, таких як LabelImg або Roboflow. Анотації повинні містити мітку класу кожного об’єкта, а також координати обмежувальної рамки [15].

Після того, як зображення анотовано, їх потрібно розділити на набори

для навчання, набори перевірки та тестові набори. Навчальний набір використовується для навчання моделі YOLOv8, а набір перевірки використовується для оцінки продуктивності моделі під час навчання та налаштування гіперпараметрів. Тестовий набір використовується для оцінки кінцевої продуктивності моделі після навчання.

Наступним кроком є створення файлу YAML, який визначає розташування наборів для навчання, перевірки та тестування, а також мітки класів для об’єктів у наборі даних. Цей файл використовується сценарієм навчання YOLOv8 для завантаження даних і налаштування процесу навчання.

Файл YAML має містити шляхи до файлу зображення та анотації, а також номер і назву класу.

Після підготовки набору даних ви можете використовувати команду yolo train для навчання моделі YOLOv8 [16]. У процесі навчання параметри моделі оптимізуються, щоб вона могла точно передбачити категорію та розташування об’єктів на зображенні. Процес навчання може тривати години або навіть дні, залежно від розміру набору даних і обраних гіперпараметрів.

Щоб точно налаштувати модель YOLOv8, існує багато гіперпараметрів, які можна налаштувати, таких як швидкість навчання, розмір партії, розпад ваги тощо. Оптимальні значення для цих гіперпараметрів залежать від конкретного набору даних і сценарію використання. Рекомендується спробувати різні гіперпараметри та оцінити продуктивність моделі на перевірочному наборі, щоб знайти найкращу комбінацію.

Таким чином, створення набору даних для навчання моделі YOLOv8 передбачає збір і анотування зображень, поділ набору даних на набори для навчання, перевірки та тестування, створення файлів YAML для налаштування процесу навчання та тонке налаштування моделі за

допомогою різних гіперпараметрів. Створення високоякісних наборів даних має вирішальне значення для досягнення високої продуктивності в задачах виявлення об’єктів і вимагає пильної уваги до деталей і точності анотацій.

2.3.2 Огляд основних джерел даних і методів побудови набору даних для розпізнавання відеоцілей

Відповідно до джерела [17], створення набору даних для розпізнавання об’єктів у відео передбачає збір і анотування зображень із обмежувальними рамками навколо цікавих об’єктів, як показано в розділі 2.3.1.

Джерело [18] також підкреслює важливість надання адекватних еталонних наборів даних для розпізнавання об’єктів у відео. Незважаючи на те, що багато наборів даних створено для конкретних завдань, деякі мають обмеження, наприклад містять лише об’єкти переднього вигляду або зміщення до певних областей зображення. Щоб створити правильний набір даних, необхідна велика кількість об’єктів з різною орієнтацією та без зміщення.

Джерело [19] містить список найкращих наборів відеоданих для розпізнавання об’єктів у 2022 році, включаючи набір даних BDD100K, який є найбільшим відкритим набором відеоданих для розпізнавання об’єктів. Цей набір даних використовується для водіння, включаючи сегментацію та відстеження кількох об’єктів, маркування зображень, виявлення дорожніх об’єктів, семантичну сегментацію, виявлення смуги руху, сегментацію доріг, сегментацію екземплярів, виявлення та відстеження кількох об’єктів, адаптацію домену та навчання симуляції.

Інші набори даних також згадуються в [18], у тому числі для розпізнавання обличчя, розпізнавання тексту, розпізнавання пішоходів, дорожніх знаків, світлофорів і виявлення об’єктів у даних дистанційного

зондування. Ці набори даних пов’язані з виявленням об’єктів і можуть використовуватися для конкретних завдань.

Таким чином, створення набору даних для розпізнавання відеооб’єктів передбачає збір і анотування зображень, розподіл їх на різні набори та запис результатів. Наявність відповідних наборів довідкових даних є критично важливою, і різні набори даних можна використовувати для різних програм.

2.4 Огляд методів оптимізації та прискорення системи відеоспостереження

Оптимізація та прискорення роботи системи відеоспостереження передбачає використання різноманітних методів. Ось деякі характеристики цих методів:

• Згорточна нейронна мережа (CNN) — це особливий тип штучної нейронної мережі (ANM), яка використовується в комп’ютерному зорі та паралельних розподілених обчисленнях для обробки великих обсягів даних, створених датчиками, і задоволення енергетичних обмежень пристроїв IoT. Штучні нейронні мережі можна використовувати для оптимізації та прискорення систем вимірювання відстані, навчаючи мережу розпізнавати та аналізувати різні об’єкти у відео. Однак навчання нейронних мереж є громіздким і трудомістким, і може зайняти дні або навіть тижні. Це обмежує використання штучних нейронних мереж у дослідницьких областях, де швидкість обчислення є надзвичайно важливою. Таким чином, необхідні достатні та вищі швидкості обчислень, щоб задовольнити потреби цих програм реального часу [20].

• Стохастичний градієнтний спуск (SGD) можна використовувати для прискорення навчання ШНМ без шкоди для точності. Оптимізація SGD є стохастичною версією алгоритму оптимізації градієнтного спуску. Це

ітераційний метод, який оновлює вагові коефіцієнти нейронної мережі, щоб мінімізувати помилку виводу. Оптимізація SGD ефективна з точки зору обчислень і може бути використана для оптимізації роботи систем вимірювання відстані у відео [20].

• Швидка згортка — ще один метод, який можна використовувати для прискорення навчання штучних нейронних мереж. Швидка згортка — це математичний метод, який зменшує кількість обчислень, необхідних для виконання операції згортання. Він заснований на алгоритмі швидкого перетворення Фур'є (ШПФ), який є ефективним алгоритмом для обчислення дискретного перетворення Фур'є (ДПФ) послідовності. Швидка згортка може оптимізувати роботу систем вимірювання відстані у відео шляхом зменшення обчислювальної складності операцій згортки [20].

• Використання паралелізму є ще одним прийомом, який можна використовувати для прискорення навчання штучних нейронних мереж. Паралелізм передбачає розподіл робочого навантаження між кількома процесорами або ядрами для одночасного виконання обчислень. Ця методика оптимізує роботу систем вимірювання відстані у відео за рахунок скорочення часу, необхідного для виконання обчислень [20].

Тому оптимізація та прискорення систем відеодальності включають CNN, оптимізацію SGD, швидку згортку, використання паралелізму та інші методи. Кожен із цих методів має свої переваги та недоліки, і вибір методу залежить від конкретних вимог програми.

2.5 Огляд методів підвищення точності вимірювання відстані до відео об'єктів

Підвищення точності вимірювання відстані до об'єктів у відеозаписах вимагає використання різноманітних методів. Ось деякі характеристики цих методів:

- Стереосистеми бачення використовують дві камери для захоплення зображень об'єкта під різними кутами, а потім використовують ці зображення для визначення розташування об'єкта в тривимірному просторі. За допомогою методів оптимізації, таких як нейронні мережі, можна підвищити точність вимірювань відстані. Наприклад, у [21] була розроблена та протестована система стереозору для виявлення загальних джерел невизначеності. Оптимізація

Використовуються для навчання нейронних мереж, отримані рівняння можуть бути реалізовані в системах стереозображення в реальному часі. Обчислювальні експерименти та порівняльний аналіз проводяться для визначення функції навчання, яка мінімізує помилку для цього підходу. Запропонований метод є загальною технікою моделювання, придатною для вирішення різноманітних завдань, що виникають у наступних областях:

Стерео система бачення. Нарешті, запропонований метод застосовується до розробленої системи стереоскопічного зображення, і виконується статистичний аналіз для перевірки отриманих покращень.

- Конфігурації ємнісного зондування матриці датчиків і машинне навчання можна використовувати для оцінки відстані та положення цільових об'єктів. [22] запропонували новий метод оцінки відстані та локалізації, який поєднує 14 конфігурацій зондування для створення 14 ємнісних датчиків для виявлення цілей. Чотири алгоритми машинного навчання для ручної оцінки відстані порівнюються з використанням потужності як вхідного вектора моделі машинного навчання. Експериментальні результати показують, що мережа радіальної базисної функції (RBF) має найкращу продуктивність: середньоквадратична помилка (RMSE), середня абсолютна помилка (MAE) і середня відносна помилка (MRE) становлять 0,59 см, 0,45 см і 5,32 %. відповідно. Результати тестування з використанням Y-подібного розташування датчиків

показують, що запропонований метод працює краще, ніж метод Y-подібного розташування датчиків, з підвищенням точності більш ніж на 98,8%, а модель BPNN досягає найвищої точності.

- Для вимірювання відстані до об'єкта можна використовувати відеокамеру. Однак важко досягти високої точності через відсутність глибинної інформації. У [24] пропонується новий метод підвищення точності вимірювання відстані на основі однієї камери огляду. Вивчається причина, чому визначення відстані з однієї камери не може досягти високої точності, і пропонується новий метод підвищення точності. Пропонований метод оцінено та порівняно з існуючими методами для демонстрації його ефективності та високої точності.

Підводячи підсумок, можна сказати, що підвищення точності визначень

Відстань до об’єктів у відео визначається за допомогою різних методів, таких як системи стереозображення, конфігурації ємнісних датчиків датчиків і машинне навчання, а також методи, засновані на використанні однієї відеокамери. Кожен із цих методів має свої переваги та недоліки, і вибір методу залежить від конкретних вимог програми.

2.5.1 Огляд основних методів підвищення точності вимірювання відстані за допомогою OpenCV і YOLOv8

У відкритому коді не згадуються конкретні способи підвищення точності визначення відстані до об’єктів у відео за допомогою OpenCV і YOLOv8. Однак існує певна дискусія, яка може допомогти досягти цієї мети.

Одним із можливих підходів є використання OpenCV для виявлення об’єктів у відеокадрах, а потім застосування методу оцінки відстані на основі виявлених об’єктів. Наприклад, у [25] каскад Хаара використовується для виявлення об’єктів, а потім оцінки відстані до об’єкта

‘

на основі його розміру на зображенні. Це можна зробити на основі відомих розмірів об’єкта або шляхом калібрування камери для оцінки розмірів об’єкта.

Інший підхід полягає у використанні визначення контурів для визначення місцезнаходження об’єктів у відеокадрах, а потім застосування методу оцінки відстані на основі контурів. У [26] розглядався метод вимірювання відстані між двома контурами у відео за допомогою OpenCV. Метод полягає в виявленні контурів у відеокадрах і обчисленні відстані між ними за допомогою формули евклідової відстані. Однак цей метод підходить лише для вимірювання відстані між двома контурами і може не підійти для більш складних ситуацій.

Загалом, підвищення точності визначення відстані до об’єктів у відео вимагає поєднання методів виявлення об’єктів, виявлення контурів і оцінки відстані. YOLOv8 — це сучасний алгоритм виявлення об’єктів, який можна використовувати в поєднанні з OpenCV для виявлення об’єктів у відеокадрах.

Після виявлення об’єкта можна застосовувати різні методи оцінки відстані залежно від конкретних вимог програми. Наприклад, для оцінки відстані до об’єктів у відео з високою точністю, як згадувалося в попередніх відповідях, можна використовувати стереосистеми бачення, конфігурації ємнісних датчиків матриці датчиків і машинне навчання, а також методи на основі камери з одним оглядом.

Таким чином, підвищення точності визначення відстані до об’єктів у відео за допомогою OpenCV і YOLOv8 вимагає поєднання методів виявлення об’єктів, визначення контурів і оцінки відстані. Вибір точного методу залежить від вимог програми, і існують різні способи досягнення високої точності.

2.6 Огляд методів підтримки визначення діапазону за допомогою

кількох камер

2.6.1 Загальні характеристики методів, які підтримують набір камер для вимірювання відстані до об’єктів у відео

Існує багато способів обчислити відстань до об'єкта за допомогою однієї стаціонарної камери, а також безліч дуже простих і лаконічних прийомів у сфері комп'ютерного зору. Однак методи обчислення відстані останнім часом стали областю досліджень, що викликає великий інтерес у робототехніці та комп’ютерному зрінні. Вимірювання відстані – це метод, який використовується в різних додатках. Наприклад, потрібно визначити відстань до системи, що знімає сцену перед автомобілем. Це дозволяє надавати інформацію про виявлені об'єкти та їх абсолютну відстань. Крім того, інформація про відстань також необхідна для самокерованих автомобілів, мультидронів і роботів. В даний час радар і лідар дозволяють визначати відстань. Проблема оцінки відстані до об’єктів у відео в даний час широко досліджується в області обробки зображень і комп’ютерного зору [8]. Одним із способів вирішення цієї проблеми є використання кількох камер. в основному

Невід’ємною частиною цього методу є багатошарова нейронна мережа Multi-DisNet, яка вивчає зв’язок між розміром обмежувальної рамки об’єкта на зображенні камери та відстанню між об’єктом і камерою [26]. Використання кількох камер забезпечує отримання необхідних даних для застосування методів дискретизації на основі відстаней перехоплення міток [27]. Багато останніх робіт почали використовувати кілька датчиків або камер для виконання різних типів завдань, таких як побудова 3D-зображення, виявлення оклюзії тощо [28].

Існує багато способів обчислити відстань до об’єкта за допомогою однієї стаціонарної камери. Одним із найефективніших методів комп’ютерного зору є метод подібності трикутника, який є найсучаснішим

методом. З його допомогою ми можемо розрахувати відстань від камери до відомого об'єкта. Цей метод заснований на принципі камери і є дуже простим і зрозумілим, якщо відомий справжній розмір об’єкта. Відстань від об’єкта до камери можна виміряти за допомогою однієї камери зі змінним кутом нахилу, покращеним за допомогою оптимізації за методом найменших квадратів. Це простий і точний метод, який дозволяє вимірювати відстань до об'єкта за допомогою камери зі змінним кутом нахилу.

Стереозйомка — це метод, який використовує дві камери для оцінки глибини точкового об’єкта з камери. Основа стереоскопічного зору подібна до тривимірного сприйняття людського зору, заснованого на тривимірному обчисленні світлових променів з кількох точок огляду. Стереозбірка — це процес пошуку точок збігу на двох зображеннях, зроблених двома різними камерами. Ми можемо взяти вікно/блок пікселів на лівому зображенні та використати його як шаблон, щоб знайти відповідне вікно такого ж розміру на правому зображенні. Існують різні методи обчислення подібності [29].

Розробляючи стереоскопічну систему, ми хочемо дуже точно виміряти відстань, оскільки це дає нам глибину. Для цього нам потрібно використовувати стереоскопічну конфігурацію з достатньо великою базовою лінією, оскільки чим більша базова лінія, тим точніше ми можемо виміряти розбіжність.

Тому існують різні методи підтримки кількох камер для визначення відстані до об’єктів у відео. Використання кількох камер надає дані, необхідні для застосування методів дискретизації відстані між мітками, тому в багатьох останніх роботах почали використовувати кілька датчиків зору або камер для виконання різних типів завдань. Стереозйомка – це метод, який використовує дві камери для оцінки відстані від камери до точкового об’єкта. Існує багато способів обчислення відстані до об’єкта за допомогою однієї стаціонарної камери, а також безліч дуже простих і

компактних методів у сфері комп’ютерного зору. Вимірювання відстані – це метод, який використовується в різноманітних програмах, і методи обчислення відстані нещодавно стали гарячою точкою досліджень у сферах робототехніки та комп’ютерного зору.

2.6.2 Огляд основних методів, які використовуються OpenCV і YOLOv8 для підтримки вимірювання відстаней до об’єктів у відео кількома камерами

За допомогою OpenCV і YOLOv8 можна використовувати кілька камер для визначення відстані до об’єктів у відео. OpenCV надає кілька методів обчислення відстані до об’єкта за допомогою однієї фіксованої камери, наприклад метод подібності трикутника, який можна використовувати для обчислення відстані від камери до відомого об’єкта. Стереозйомка — ще один метод, який можна використовувати для оцінки глибини точкового об’єкта з камери за допомогою двох камер.

Щоб реалізувати виявлення об’єктів і вимірювання відстані за допомогою YOLOv8, існує сценарій під назвою object_detection.py у репозиторії виявлення об’єктів і вимірювання відстані на GitHub. Сценарій використовує YOLOv3 для виявлення об’єктів у відеокадрах, а потім обчислює відстань об’єкта від камери, використовуючи інформацію про глибину, яка використовується камерою для малювання обмежувальних рамок для визначення місцезнаходження об’єкта [30]. Відстань вимірюється шляхом обчислення ширини та висоти обмежувальної рамки, яка змінюється залежно від відстані об’єкта до камери. Відстань можна розрахувати за такою формулою:

$$distance = (2 \times 3{,}14 \times 180) / (w + h \times 360) \times 1000 + 3$$

де w і h — ширина і висота обмежувальної рамки відповідно. Ця формула враховує заломлення зображення під час його проходження через об’єктив камери, а також різницю в розмірах об’єктів на зображенні.

Ще один спосіб підтримувати кілька камер у визначенні відстані до об’єктів у відео – це використовувати багатошарову нейронну мережу під назвою Multi-DisNet, яка вивчає зв’язок між розміром обмежувальної рамки об’єктів на зображенні камери та відстанню між об’єктами. . об’єктів і камер [32]. Цей метод корисний для отримання даних, необхідних для застосування методу дискретизації за відстанню перехоплення мітки.

Тому існує кілька способів підтримки кількох камер для визначення відстані до об’єктів у відео за допомогою OpenCV і YOLOv8. OpenCV надає методи обчислення відстані до об’єкта за допомогою однієї фіксованої камери, тоді як стереобачення можна використовувати для оцінки глибини точкового об’єкта з камери за допомогою двох камер. У репозиторії виявлення об’єктів і вимірювання відстані на GitHub є сценарій, який використовує YOLOv3 для виявлення об’єктів у відеокадрах і обчислення відстані до об’єкта від камери, визначаючи місце розташування об’єкта за допомогою інформації про глибину, яка використовується камерою для малювання обмежувальних рамок полягає у використанні багатошарової нейронної мережі Multi-DisNet, яка вивчає зв’язок між розміром обмежувальної рамки об’єкта на зображенні камери та відстанню між об’єктом і камерою.

2.7 Огляд методу автоматичного калібрування системи вимірювання відстані до об'єкта

У статті [33] пропонується повністю автоматична та гнучка процедура калібрування стаціонарних багатокамерних систем для тривимірного спостереження динамічних подій на основі послідовностей зображень. Метод заснований на нерухомій камері та рухомій цілі, використовуючи переваги послідовного зображення більшості твердотільних камер. Метод заснований на відстеженні одного, легко

виявленого маркера через послідовність зображень з кількох попередньо відкаліброваних камер, таким чином уникаючи необхідності ідентифікації гомологічних ознак для встановлення відповідності між кількома видами. У більш просунутих версіях еталонна смуга відомої довжини переміщується в просторі об'єкта, а задача ідентифікації ознак і встановлення багаторакурсної відповідності зводиться до супроводу двох цілей. Постійна довжина опорної лінії використовується як додаткова інформація про геометричні обмеження під час самокалібрування променя, що значно покращує рішення та дозволяє повноцінно калібрувати кожну камеру багатокамерної системи, включаючи параметри внутрішньої орієнтації.

Методи «Перемістити точку» та «Перемістити опорну лінію» прості та зрозумілі для реалізації, але мають певні обмеження. Якщо зовнішню орієнтацію та параметри кривизни лінзи можна визначити, внутрішню орієнтацію камери неможливо відновити на основі інформації з однієї точки зміщення. Зіставлення також зводиться до виявлення або відстеження двох об’єктів або цілей, що робить його менш придатним для самокалібрування. Метод «рухомої еталонної лінії» можна використовувати для визначення внутрішньої орієнтації кількох камер, але детальний аналіз цього методу відсутній, а також схема калібрування, включаючи специфікації ідеальної кількості, розташування та орієнтації послідовних еталонних камер. лінії спостережень, які слід застосовувати для калібрування систем.

Інші методи автоматичного калібрування системи для визначення відстані до об’єктів у відео включають використання калібрувального шаблону, наприклад шахової дошки або сітки, які знімаються камерою під різними кутами для оцінки внутрішніх і зовнішніх параметрів об’єкта. камера. OpenCV надає функції для калібрування камер, включаючи calibrateCamera() і stereoCalibrate(), які можна використовувати для калібрування окремих камер і стереокамер відповідно. Ці функції оцінюють

матрицю камери, коефіцієнти спотворення та вектори обертання та трансляції та можуть бути використані для декомпозиції та корекції зображення, а також обчислення тривимірного положення точок у сцені [34].

2.8 Загальна характеристика методів автоматичного калібрування систем вимірювання відстані до відеооб'єктів

Одним із найпоширеніших способів автоматичного калібрування системи за допомогою OpenCV для визначення відстані до об’єктів у відео є калібрування камери в шаховому порядку. Цей метод усуває радіальні та тангенціальні спотворення, які можуть вплинути на оригінальне зображення і, таким чином, на оригінальні вимірювання об’єктів на зображенні [35]. OpenCV надає функції калібрування камери, включаючи calibrateCamera() і stereoCalibrate(), які можна використовувати для калібрування окремих камер і стереокамер відповідно. Ці функції оцінюють матрицю камери, коефіцієнти спотворення та вектори обертання та трансляції та можуть використовуватися для спотворення та виправлення зображень, а також для розрахунку 3D-положень точок у сцені.

Інший спосіб автоматичного калібрування системи визначення відстані до об’єктів у відео полягає у використанні еталонного об’єкта відомих розмірів, наприклад шахової дошки або сітки, та оцінки внутрішніх і зовнішніх параметрів камери на основі зображень об'єкт розглядається з різних кутів. Цей метод подібний до методу калібрування камери в шаховому порядку, але замість шахової дошки використовуються об’єкти відомого розміру. Після оцінки внутрішніх і зовнішніх параметрів можна розрахувати відстань до об’єктів на відео за допомогою таких методів, як подібність трикутника або стереобачення.

Таким чином, основний метод автоматичного калібрування систем,

що використовують OpenCV і YOLOv8 для визначення відстані до об’єктів на відео, полягає в шаховому режимі камери та використанні еталонного об’єкта відомого розміру. OpenCV надає функції калібрування камери, які можна використовувати для оцінки матриці камери, коефіцієнтів спотворення та векторів обертання та трансляції. Після оцінки цих параметрів відстані до об’єктів у відео можна обчислити за допомогою різних методів, таких як методи подібності трикутників і стереобачення.

2.9 Огляд фреймворків і бібліотек для розробки веб-додатків з можливістю обробки відеоданих

2.9.1 Вивчіть можливість використання Flask для розробки веб-додатків із можливістю обробки відеоданих

Flask — це легкий фреймворк веб-додатків WSGI, призначений для швидкого й легкого запуску та здатний масштабуватися до складних додатків [36]. Flask — це мікрофреймворк для швидкого створення простих веб-додатків. Він включає підтримку шаблонів Jinja, обробки запитів і сигналів додатків. У порівнянні з претензійним фреймворком Django, Flask

пропонує більшу гнучкість, що робить його більш придатним для програмістів з великим досвідом кодування або тих, кому потрібен більший контроль над дизайном програми [37].

Flask можна використовувати для розробки веб-додатків, здатних обробляти відеодані. Наприклад, у посібнику «Потокове відео у веб-браузері за допомогою Flask і OpenCV» автори демонструють, як транслювати відео з веб-камери у веб-браузері за допомогою Flask і OpenCV [37]. Інший навчальний посібник із розгортання серверної програми обробки відео на Python за допомогою Flask і OpenCV містить код для обробки відеокадрів за допомогою OpenCV і відображення обробленого відео у веб-браузері за допомогою Flask [38].

Flask також можна використовувати для створення веб-додатків, які дозволяють користувачам завантажувати та обробляти відеофайли. Наприклад, у цьому відеопосібнику YouTube про потокове передавання через веб-камеру у веб-платформі Flask автор демонструє, як використовувати Flask для створення веб-програми, яка дозволяє користувачам завантажувати відеофайли, які потім обробляються та відображаються у веб-браузері за допомогою наступного: команди Відеофайли OpenCV і Flask [39].

Таким чином, Flask є легкою структурою веб-додатків, яку можна використовувати для розробки веб-додатків із підтримкою відео. Flask можна використовувати для потокової передачі відео з веб-камери у веб-браузері, обробки відеокадрів за допомогою OpenCV і створення веб-додатків, які дозволяють користувачам завантажувати та обробляти відеофайли.

2.9.2 Розгляньте можливість використання Flask для створення веб-додатків, які використовують нейронні мережі

Flask можна використовувати для створення веб-додатків, які

використовують штучний інтелект для розпізнавання відеозображень. Насправді Flask можна використовувати разом із бібліотеками ШІ, такими як OpenCV, для створення таких програм. Наприклад, у навчальному посібнику про те, як використовувати Flask для відображення потокового відео з веб-камери, автори демонструють, як захоплювати та відображати відео у веб-браузері за допомогою Flask і OpenCV [40]. В іншому навчальному посібнику зі створення веб-програми машинного навчання за допомогою Flask описано, як використовувати Flask для створення веб-програми, яка дозволяє користувачам завантажувати зображення, а потім використовувати модель машинного навчання для класифікації зображень [41].

Бібліотеки штучного інтелекту, такі як OpenCV, можна використовувати для інтеграції штучного інтелекту розпізнавання відео у веб-програми Flask. Наприклад, у підручнику з інтеграції API Clarifai у програму Flask автор демонструє, як використовувати Flask і API Clarifai для розпізнавання зображень їжі [42]. Подібним чином, використовуючи Flask і модель машинного навчання, було створено веб-додаток для прогнозування зображень цифр жестів, як описано в підручнику зі створення програми Flask для розпізнавання зображень [43].

У цьому розділі вивчаються та розглядаються різні методи, техніки та бібліотеки для систем, що використовуються для вимірювання відстаней до об’єктів у відео.

Спочатку надається огляд бібліотеки OpenCV та її функціональних можливостей, а потім розглядаються загальні характеристики OpenCV та її використання в комп’ютерному зорі. Далі розглядаються основні функції та алгоритми OpenCV для обробки відео.

Далі характеризується мережева архітектура YOLOv8 та окреслюються її особливості та переваги для обробки відео. Це дослідження містить загальний опис методу побудови набору даних для навчання мережі

YOLOv8, а також огляд основних джерел даних і методів побудови набору даних для розпізнавання відеооб’єктів.

Потім розглядаються методи оптимізації та прискорення роботи систем відеоспостереження, а також підвищення точності визначення відстані до об’єктів на відео.

Також розглядаються методи підтримки кількох камер для визначення відстані до об’єктів у відео, включаючи загальний опис цих методів і огляд основних методів підтримки кількох камер за допомогою OpenCV і YOLOv8.

Надається огляд методів систематичного автоматичного калібрування для визначення відстаней до об’єктів на відео, включаючи загальний опис цих методів та їхні переваги та недоліки.

Нарешті, розглядаються фреймворки та бібліотеки для розробки веб-додатків із можливістю обробки відеоданих. Зокрема, досліджується можливість розробки таких веб-додатків за допомогою Flask, а також можливість використання Flask для створення веб-додатків за допомогою людей інтелект розпізнавання образів відео. Крім того, визначено переваги цього фреймворку перед іншими фреймворками, що принципово покращило фреймворк, обраний для виконання дипломного проекту.

3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Вибір алгоритму виявлення цілі

Вибір відповідного алгоритму виявлення об'єктів є критичним кроком у задачах комп'ютерного зору. Одним із найсучасніших та ефективніших алгоритмів виявлення об’єктів є YOLOv8. Алгоритм вражає здатністю працювати в реальному часі та досягати високої точності розрізнення об’єктів.

YOLOv8 (You Only Look Once, версія 8) — одна з останніх розробок у серії алгоритмів YOLO. Він використовує глибокі нейронні мережі,

зокрема згорточні нейронні мережі, для виявлення та класифікації об’єктів на зображеннях або відео. Однією з головних переваг YOLOv8 є його швидкість і здатність працювати в реальному часі. Це робить його особливо привабливим для програм, де важлива швидка обробка відеоданих або потоків зображень. (Рисунок 3.1).

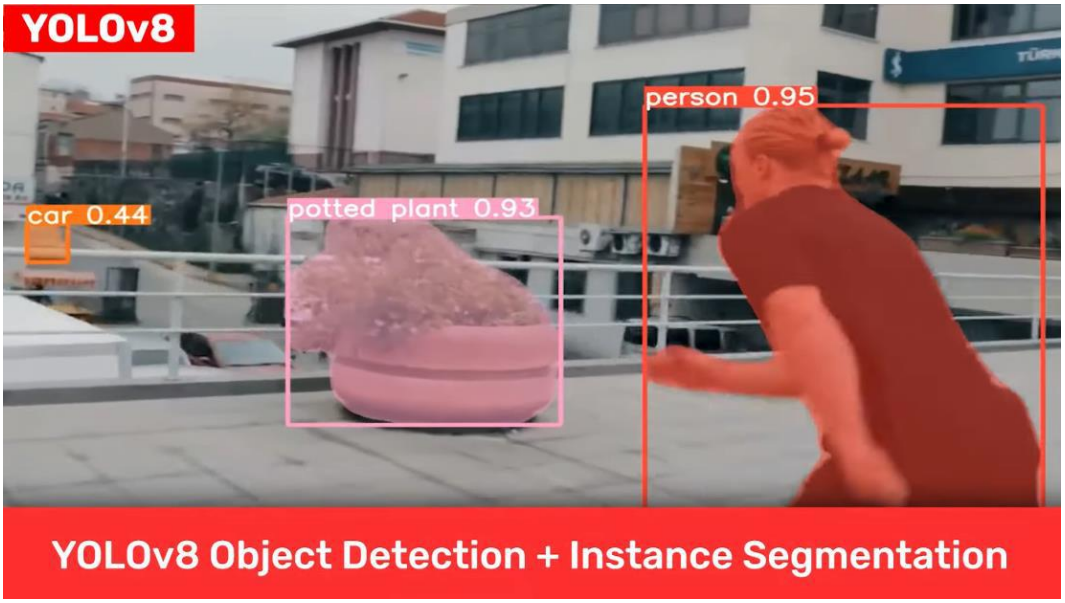


Рисунок 3.1-Демонстрація роботи алгоритму YOLOv8[23]

Найбільш вражаючою особливістю YOLOv8 є його висока точність виявлення об'єктів. Алгоритм здатний надійно та точно ідентифікувати різні категорії об’єктів незалежно від їх розміру та форми. Він демонструє високий ступінь стійкості до змін середовища зображення, таких як зміна освітлення, накладання об’єктів або зміна кутів огляду. Це робить його універсальним і надійним інструментом для розпізнавання об’єктів у різноманітних сценаріях та умовах.

Крім того, YOLOv8 — це алгоритм, здатний ідентифікувати багато об’єктів у кадрі зображення чи відео одночасно. Це дозволяє виявляти та диференціювати об’єкти різних категорій, надаючи повний контекст і повну інформацію про зображення. Це особливо корисно для сценаріїв, де є кілька

об’єктів, і для завдань, які потребують визначення та розуміння взаємодії між об’єктами. (Рисунок 3.2).

Рисунок 3.2 - Результати YOLO [13] ідентифікації людей на зображеннях

Тому в якості алгоритму виявлення цілей доцільно вибрати YOLOv8 через його сучасність, здатність працювати в режимі реального часу та високу точність виявлення цілей. Алгоритм є потужним і надійним інструментом у комп’ютерному зорі, і його можна використовувати для різних завдань: від безпілотних транспортних засобів до систем відеоспостереження та доповненої реальності.

3.2 Вибір методу вимірювання відстані

Виявлення та визначення відстані є одним з ключових завдань у сфері комп'ютерного зору. Процес включає ідентифікацію та локалізацію об’єктів на зображеннях або відео. За останні роки було розроблено багато алгоритмів і методів для вирішення цієї проблеми, і вибір відповідного алгоритму може бути складним завданням.

Одним із можливих способів визначення відстані до об’єкта є використання інформації про пікселі та фокусну відстань камери. Метод заснований на використанні геометричних властивостей зображення та

фізичних властивостей об'єкта. У використання цього підходу є кілька переваг, зокрема простота впровадження та можливість використовувати одну камеру в режимі реального часу. (Рисунок 3.3)

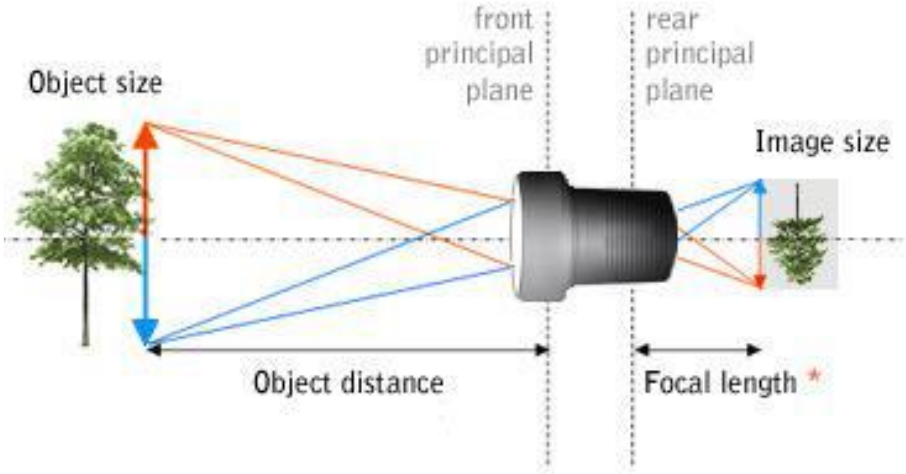


Рисунок 3.3 – Демонстрація алгоритму на основі фокусної відстані [35]

Значення пікселів на зображенні можна використовувати для визначення розташування та форми об'єктів. Цього можна досягти шляхом аналізу інтенсивності пікселів та їх кореляції. Наприклад, різниця в яскравості між об’єктом і фоном може вказувати на присутність об’єкта на зображенні. Ці методи можна реалізувати за допомогою фільтрів, порогових значень або відстеження контурів. Крім того, моделі на основі нейронних мереж, такі як YOLO, можна використовувати для визначення розмірів об’єктів і, таким чином, їхнього положення на живих зображеннях без затримки. (Рисунок 3.4).

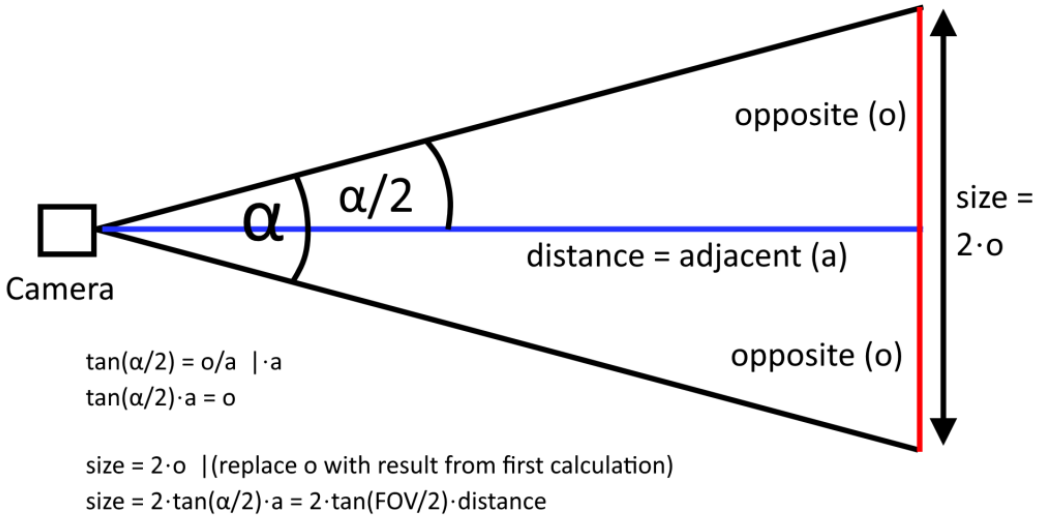


Рисунок 3.4 – Демонстрація алгоритму вимірювання відстані на основі пікселів

Крім того, фокусну відстань камери можна використовувати для визначення відстані до об’єктів на зображенні. Це можна зробити, проаналізувавши розміри об’єктів на зображенні та їхню відстань від камери. Знання фокусної відстані дозволяє виконувати визначення глибини об’єктів, що важливо в багатьох сферах, таких як робототехніка, автономні транспортні засоби та віртуальна реальність.

Поєднуючи піксельну інформацію з фокусною відстанню камери, можна отримати точнішу інформацію про розташування та властивості об’єкта. Знання відстані до об’єктів дозволяє враховувати їх просторове співвідношення, що корисно при визначенні орієнтації та масштабу об’єкта.

Існує кілька причин для вибору алгоритму виявлення об’єктів на основі пікселів і фокусної відстані. По-перше, метод простий у реалізації, оскільки він заснований на стандартних методах аналізу зображення та розрахунку фокусної відстані камери. Це означає, що його можна легко використовувати та не потребує складних обчислювальних ресурсів чи спеціального обладнання.

По-друге, методи на основі пікселів і фокусної відстані можна успішно використовувати з однією камерою в режимі реального часу. Це означає, що його можна застосовувати в реальному часі без необхідності обробляти або аналізувати записані відеодані. Це робить його ефективним і зручним у використанні в різноманітних програмах, де важлива негайна реакція на об’єкти.

Враховуючи простоту реалізації та можливість використання однієї камери в полі, методи на основі пікселів і фокусної відстані є привабливим варіантом для виявлення об’єктів.

3.3 Вибір даних

Вибір правильних даних є критичним етапом у розробці системи виявлення об’єктів. Виберіть відкритий набір даних Kaggle, щоб розробити систему виявлення об’єктів зображення. Враховуючи опис набору даних, наданий на веб-сайті, наданий набір даних має кілька переваг і може бути відповідним вибором для розробки детекторів об’єктів.

Набір даних був створений лабораторією AIoT Національного тайванського університету науки і технологій за допомогою фотографій високої роздільної здатності, зроблених дронами, оснащеними камерами 4K. Зображення високої роздільної здатності можуть значно покращити здатність сучасних детекторів предметів ідентифікувати невеликі предмети. Принадність цього набору даних полягає в тому, що він містить справжні фотографії, зроблені з дронів, а не просто збільшені зображення. Це дає можливість отримати більш реалістичні дані для навчання моделей виявлення об’єктів. (Рисунок 3.5).

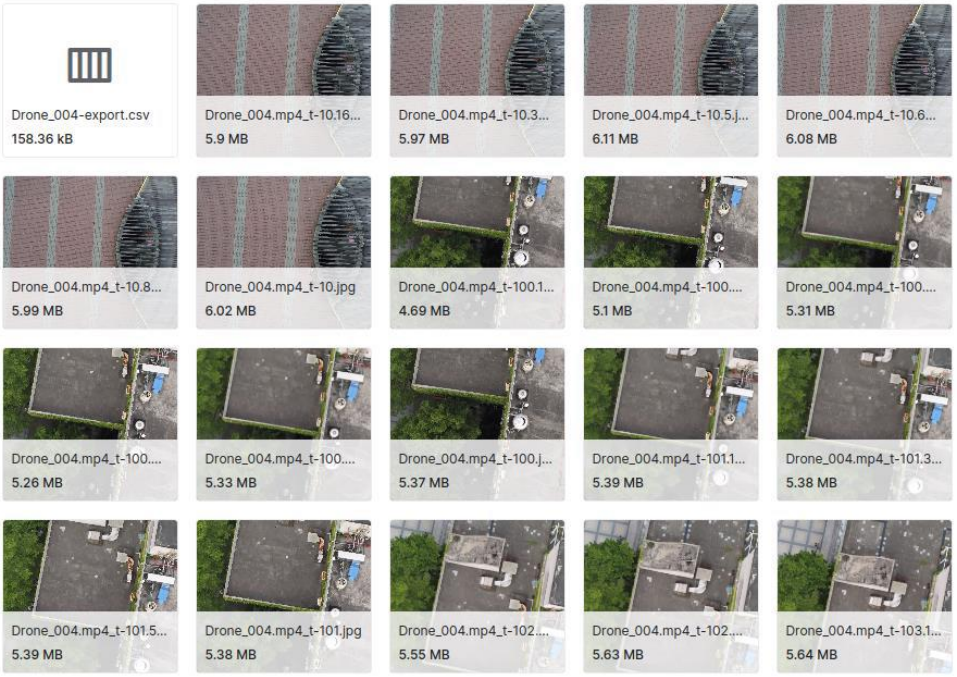


Рисунок 3.5 – Перегляд зображень у наборі даних

Як видно із зображення вище, навчальне зображення — це набір відеокадрів, розділених певним інтервалом. Кожне зображення займає багато місця в пам'яті, в середньому 5,4 Мб. Розміри зображення становлять 3840x2160 пікселів, що відповідає розмірам зображення 4K. (Рисунок 3.6).

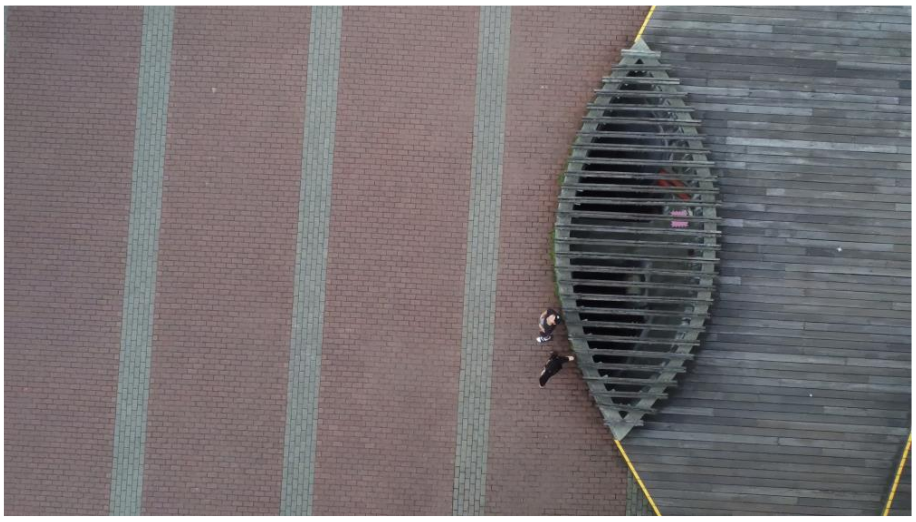


Рисунок 3.6 – Одне із зображень, наданих у наборі даних.

‘

Цей набір даних містить чотири набори даних, зібраних у різних ситуаціях. Об’єкти в цих колекціях позначені тегами. Маркери включають людей, які ходять, стоять, їздять, дивляться на свої телефони тощо. Це маркування надає важливі дані про різноманітність об’єктів, які допоможуть моделі точніше ідентифікувати людей, оскільки додавання нових категорій покращує можливості розпізнавання моделей YOLO. (Рисунок 3.7).

Рисунок 3.7 - Вид зображення з мітками

Дані зображення для кожного набору даних зберігаються у форматі CSV. Ці файли CSV містять інформацію про розташування об’єктів і посилання на зображення, до яких вони належать. Це полегшує впорядкування та використання даних для навчання моделей. Координати об'єкта (xmin, ymin, xmax, ymax) і мітка (label) вказуються в файлі CSV, що дозволяє точно визначити розмір і категорію об'єкта. (Рисунок 3.8).

image	xmin	ymin	xmax	ymax	label
Drone_005.mp4_t-12.833333	2195.703128246235	159.44943494639233	2343.904368769805	299.991714358157	walk
Drone_005.mp4_t-12.833333	2195.703128246235	159.44943494639233	2343.904368769805	299.991714358157	push
Drone_005.mp4_t-12.833333	2485.488565488675	220.78817733990147	2626.528066528067	367.0935960591133	walk
Drone_005.mp4_t-12.666667	2219.653232196073	154.12923790205735	2367.854472720842	294.6715173138205	walk
Drone_005.mp4_t-12.666667	2219.653232196073	154.12923790205735	2367.854472720842	294.6715173138205	push
Drone_005.mp4_t-12.666667	2509.438669438712	207.48768472906403	2650.478170478171	353.7931034482758	walk
Drone_005.mp4_t-12.5	2243.603336146411	135.50854824688494	2391.804576670188	276.0508276586496	walk
Drone_005.mp4_t-12.5	2243.603336146411	135.50854824688494	2391.804576670188	276.0508276586496	push
Drone_005.mp4_t-12.5	2544.033264033654	186.20689655172413	2685.072765072765	332.5123152709359	walk
Drone_005.mp4_t-12.333333	2259.570072113147	114.22776006954504	2407.771312636923 5	254.7700394813097 4	walk
Drone_005.mp4_t-12.333333	2259.570072113147	114.22776006954504	2407.771312636923 5	254.7700394813097 4	push

Рисунок 3.8 – Перегляд файлу даних зображення

Крім того, набір даних розділений на тестовий і навчальний набори, що дуже важливо для оцінки та налаштування моделей виявлення об’єктів. Розділення папок для навчання та тестування дає змогу належним чином використовувати дані для навчання та оцінки моделі, забезпечуючи належну перевірку її продуктивності на незалежних наборах даних. (Рисунок 3.9)

‘

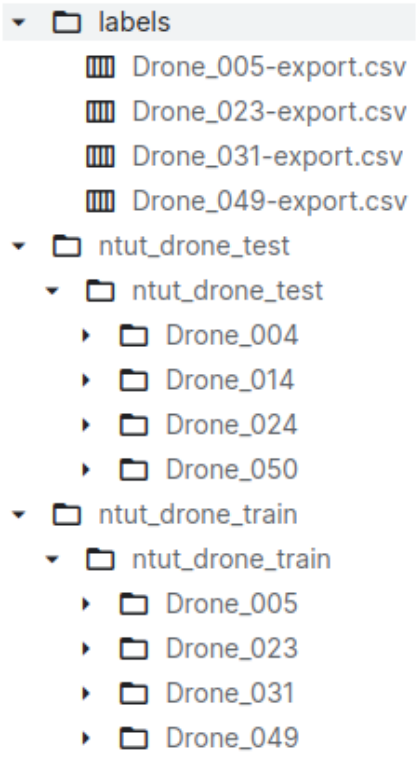


Рисунок 3.9 – Структура каталогу набору даних

Таким чином, наведений вище набір даних є хорошим вибором для розробки систем виявлення об’єктів, оскільки він містить зображення з високою роздільною здатністю, різні немарковані об’єкти та пов’язані дані для навчання та тестування моделі.

3.4 Обробка даних

Для навчання обраної моделі дані необхідно конвертувати в спеціальний формат. Формат даних виявлення YOLO вимагає двох різних типів файлів для навчання: файли зображень і файли міток.

Файли зображень мають такі властивості:

• Кожне зображення повинно мати власний унікальний ідентифікатор, наприклад ім'я файлу або шлях до зображення.

• Зображення можуть бути в різних форматах, наприклад JPEG або

PNG. Файли міток мають такі характеристики:

• Кожен рядок у файлі мітки відповідає об’єкту на зображенні та має такий формат: <клас> <центр x> <центр y> <ширина> <висота>

o <клас> - мітка класу об'єкта, яка може бути числовим або текстовим ім'ям класу.

o <x-center>, <y-center> - відносні координати центру об'єкта, де (0, 0) - верхній лівий кут, а (1, 1) - нижній правий кут

зображення.

o <width>, <height> - відносні розміри об'єкта на основі ширини та висоти зображення.

• Файл міток може містити мітки для кількох об’єктів на одному зображенні. Кожна лінія представляє окремий об’єкт.

Таблиця 3.1 - Приклад файла з мітками

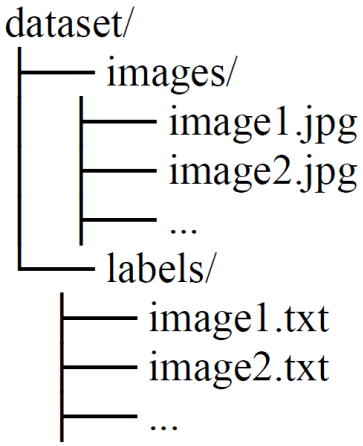
0	0.45	0.60	0.30	0.40
1	0.75	0.30	0.60	0.55
2	0.20	0.70	0.25	0.35

У наведеному вище прикладі перший рядок вказує на те, що зображення має об’єкт класу 0 із центром у координатах 0,45 і 0,60, шириною 0,30 і висотою 0,40. Другий рядок вказує на наявність об’єкта класу 1 з центром у координатах 0,75 і 0,30, шириною 0,60 і висотою 0,55. Третій рядок вказує на наявність об’єкта класу 2 із центрами в координатах 0,20 і 0,70, шириною 0,25 і висотою 0,35.

Дані у файлі мають бути нормалізовані таким чином, щоб вони були між 0 і 1. Це зроблено для того, щоб алгоритм міг обробляти зображення будь-якого розміру.

Набір даних моделі YOLO повинен мати певну структуру каталогів

для навчання:



Де:

• набір даних - основна папка, що містить усі дані, які використовуються для навчання або тестування моделі YOLO.

• Зображення - папка, де зберігаються зображення. Кожне зображення повинно мати власний унікальний ідентифікатор, наприклад ім’я файлу або шлях до зображення.

• labels - папка для зберігання файлів етикеток для кожного зображення. Кожен файл мітки відповідає певному зображенню та містить інформацію про виявлені на зображенні об’єкти у форматі, який підходить для моделі YOLO. Файли тегів зазвичай мають той самий ідентифікатор, що й відповідне зображення, але мають розширення .txt або інше відповідне розширення.

Цю структуру каталогу можна повторити для наборів тестів і перевірки моделей.

Крім того, вам потрібно створити файл YAML, що містить дані класу та папки, щоб навчити модель. Файл повинен мати таку структуру:

train: шлях/до/набору/даних/train.txt
val: шлях/до/набору/даних/val.txt

nc: кількість_класів
names: [клас_1, клас_2, ..., клас_nc]

Виходячи з наведеної вище структури, файл data.yaml для завдання виглядає так:

```
path: /kaggle/working/data
train: /kaggle/working/data/train
val: /kaggle/working/data/test
nc: 9
names:
  0: walk
  1: push
  2: block50
  3: block75
  4: stand
  5: block25
  6: watchphone
  7: riding
  8: sit
```

Як видно з рисунка вище, для навчання моделі використовуються два набори даних, включаючи дані навчання та дані перевірки. Модель пройшла навчання за 9 категоріями, включаючи людей, які ходять, штовхаються, стоять, дивляться в мобільні телефони, водять, сидять тощо.

3.5 Модельне навчання

Модель тренували на три епохи. Оскільки розмір набору зображень

становить 4K, а кількість досить велика, додаткове тестування оптимальної кількості епох не проводилося. За допомогою вбудованого графічного процесора Kaggle навчання можна виконувати в хмарі швидше, ніж на вашому власному пристрої, і не перевантажуючи свій комп’ютер. Таблиця результатів навчання виглядає наступним чином:

Таблиця 3.2 – Ілюстрація результатів тренування моделі

epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)	val/box_loss	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1	lr/pg2
0	1.93371	3.90021	1.4144	0.81632	0.14433	0.61004	0.26059	1.7639	4.2003	1.3285	0.070323	0.003295	0.003295
1	1.8068	2.0226	1.3909	0.78823	0.7226	0.76159	0.33305	1.864	2.2885	1.5328	0.038134	0.004447	0.004447
2	1.7422	1.7542	1.3509	0.78004	0.74087	0.74862	0.3259	1.8216	1.8807	1.4815	0.037462	0.003378	0.003378

Додатково створено табличну візуалізацію за допомогою діаграм:

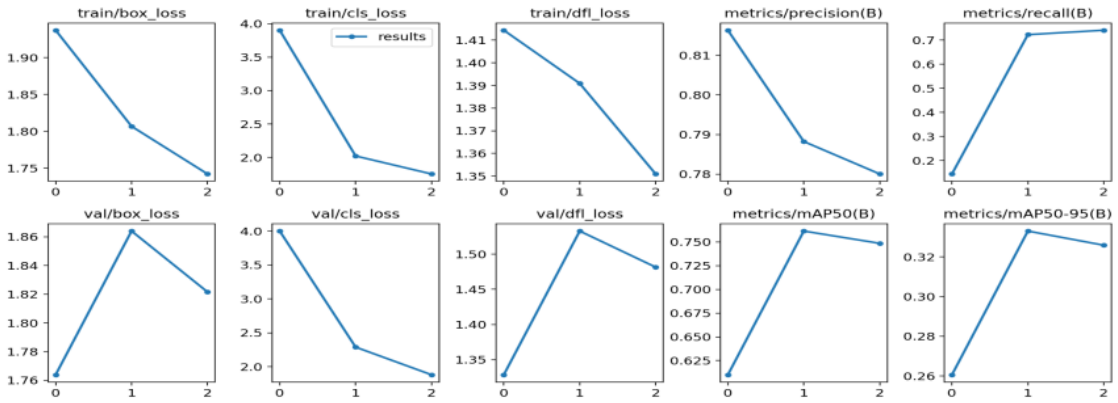


Рисунок 3.10 – Візуалізація таблиці

Як видно з таблиці та рисунка вище, обсяг навчання трьох епох є цілком ідеальним, оскільки графік точності починає знижуватися після 2 епох.

3.6 Оцінка результату моделі

Після навчання моделі було створено декілька зображень, які демонструють здатність моделі виявляти людей на зображеннях:

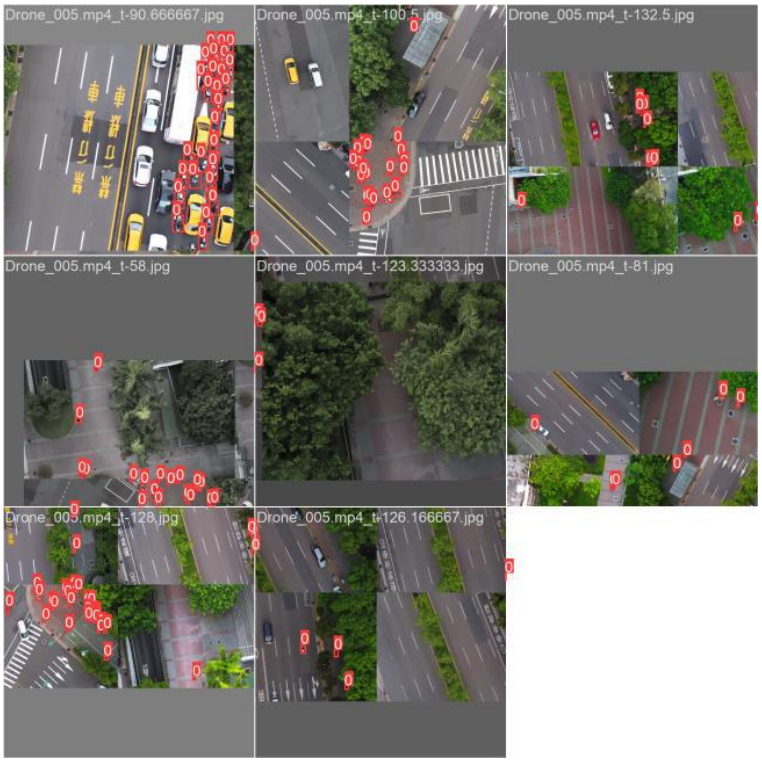


Рисунок 3.11 - Результати тренування Batch0

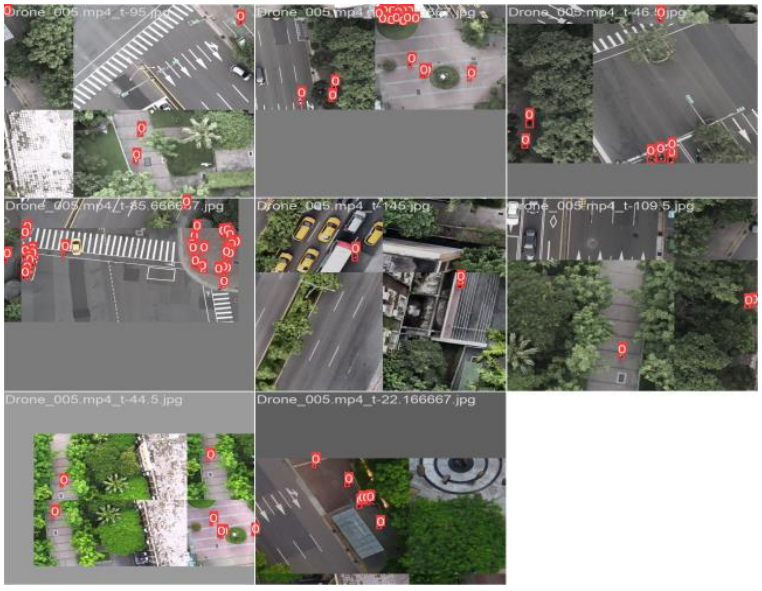


Рисунок 3.12 - Результати тренування Batch1

‘

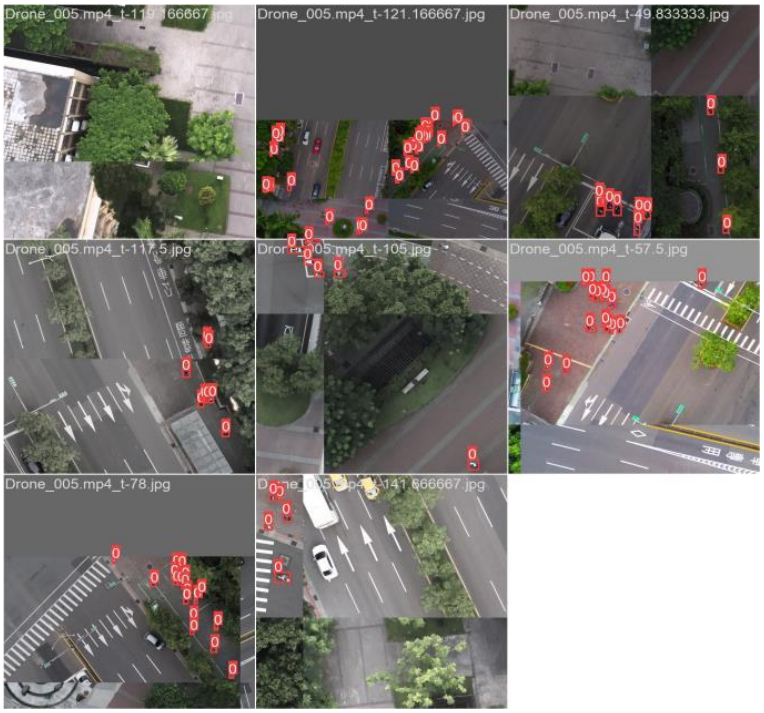


Рисунок 3.11 - Результати тренування Batch2

Як ви можете бачити на зображенні вище, модель дуже добре справляється з виявленням людей на зображеннях. Враховуючи невеликий розмір символів у даному кадрі, якість результатів справді висока. Однак важливо зазначити, що наведене вище зображення є результатом роботи моделі з навчальними даними. Для візуальної перевірки моделі було створено три додаткові набори зображень на основі даних перевірки:

Рисунок 3.12 - Правдиві мітки даних на val_batch2

Рисунок 3.13 - Передбачені мітки даних на val_batch2

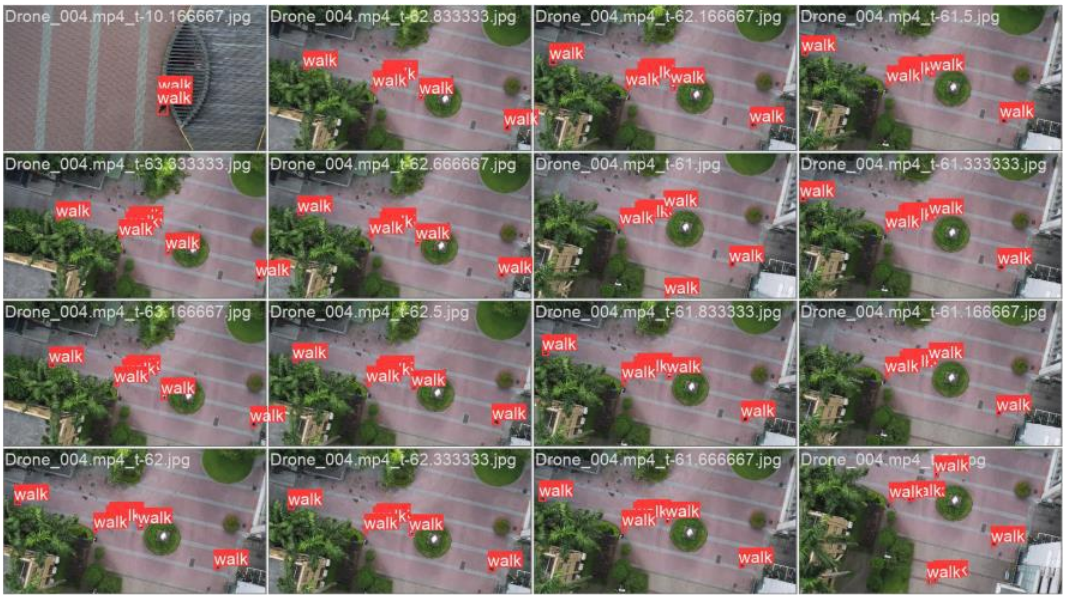


Рисунок 3.13 - Правдиві мітки даних на val_batch1

Рисунок 3.14 - Передбачені мітки даних на val_batch1

Як видно з отриманих зображень, модель дуже успішно виявляє людей навіть на такій відстані. Однак існують певні проблеми з виявленням зайвих об’єктів і позначенням їх як людей. Це можна вирішити шляхом додаткового навчання моделі на нових типах даних.

‘

3.7 Комбінація алгоритмів і моделей

Тому в результаті розробки сформувалася модель, яка може визначати місцезнаходження людей через камери дронів. Наступним кроком впровадження системи є розробка алгоритму визначення відстані до людини.

Алгоритм використовуватиме дані про ширину рамки навколо людини, фактичний розмір позначеної області та фокусну відстань дрона. Фокусна відстань камери не може бути змінена, тому вона завжди постійна. Ширина рамки навколо людини вимірюється в пікселях, і це буде наша змінна. Фактичні розміри виділеної області становлять від 40 до 35 см, ці гіперпараметри пізніше можна налаштувати для конкретного пристрою зчитування зображення, а фактичні розміри об’єкта також можуть бути ще одним параметром нейронної мережі, який дозволить здійснювати процес визначення. Відстань до повністю автоматизованого об'єкта за наявності відповідних даних.

Відстань до об'єкта розраховується за формулою (3.1):

$$distance = (focal\_length * object\_width) / (w * img\_w) \quad 3.1$$

де focal_le'ngth — фокусна відстань пристрою, object_width — фактичний розмір вибраної області, w — кількість пікселів у вибраній області (xmax-xmin), а img_w — кількість пікселів у всьому зображенні.

Ця формула дозволяє визначити відстань між камерою та об'єктом, використовуючи вхідні дані нейронної мережі та дані про об'єкт і дані про камеру.

Розроблені функції включають можливість отримувати зображення та дані про об’єкти та камери, виконувати виявлення об’єктів на відео, виконувати обчислення та виводити результати.

3.8 Оцінка продуктивності системи

Результатом розробки стала система, яка може визначати відстань до людини за відеокадрами, знятими з літаючого дрона.

Модель перевіряли на тренувальних даних, деякі параметри (наприклад, фокусна відстань) були невідомі на початку, оскільки модель дрона не була вказана в наборі даних, а потім були обрані методом проб і помилок.

Результати моделі виглядають так:

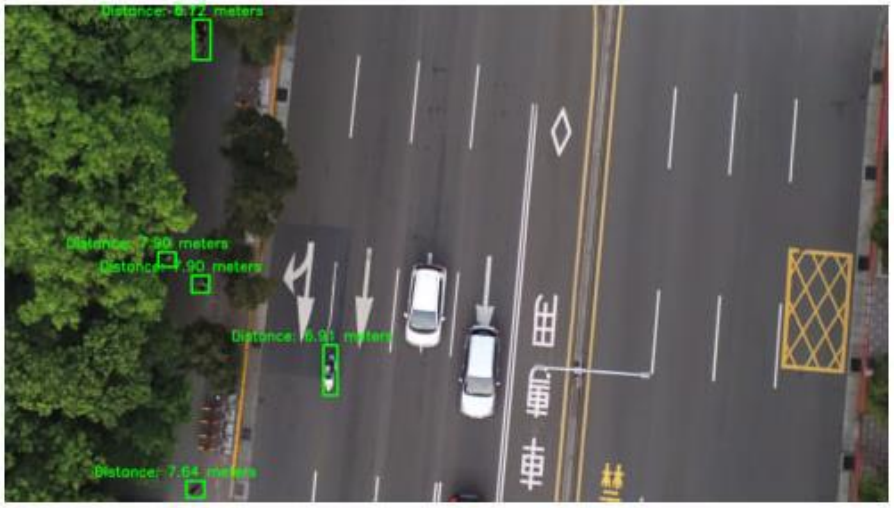


Рисунок 3.15 - Результати розробленої моделі

Як ви можете бачити на зображенні вище, модель може дуже добре виявляти об’єкти на відео, але її точність значною мірою залежить від точності виявлення кадрів, що оточують об’єкт на зображенні. Щоб підвищити продуктивність, до моделі слід додати нові дані, а для більш точного визначення розміру об’єктів у пікселях можна використовувати методи сегментації. Нейронні мережі також можна використовувати для додаткового прогнозування розмірів об’єктів, що може значно покращити результати роботи.

3.9 Розробка інтерфейсу

У цій роботі бібліотека Python Flask використовується для реалізації зручного інтерфейсу для взаємодії з моделлю. Функціональні можливості цієї бібліотеки було описано в попередніх розділах, надано кілька джерел, які демонструють можливості інструменту для роботи з нейронними мережами та його здатність і легкість інтеграції з нейронними мережами, які надає Flask.

Для користувачів створено дві сторінки:

• Сторінка, яка надає можливість вибрати файл і завантажити його, тим самим запускаючи нейронну мережу

• Сторінка відображення результатів, яка дозволяє користувачам завантажувати відео для перегляду результатів

Розроблена сторінка виглядає так:

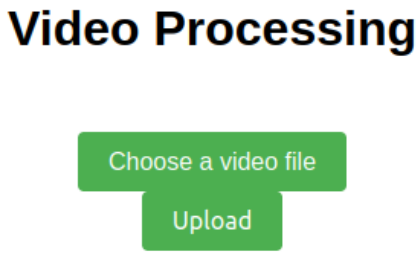


Рисунок 3.16 - Панель взаємодії із системою на першій сторінці

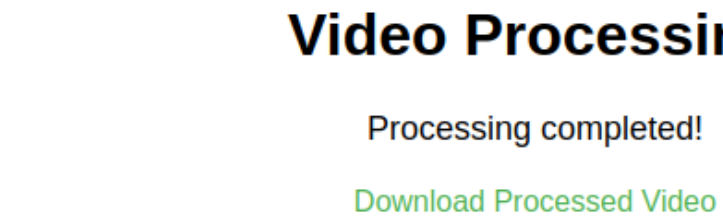


Рисунок 3.17 – Сторінка виведення моделі

Як видно на зображенні вище, дизайн інтерфейсу досить мінімалістичний, але забезпечує безперебійну та комфортну роботу на розробленій системі тестування роботи та попереднього перегляду результатів.

Код, наданий для взаємодії з інтерфейсом, завантажує дані в папку завантаження, потім запускає кадри один за одним у модель і записує їх у відео у вихідній папці. Каталог ваг містить ваги моделі для зручного налаштування системи. (Рисунок 3.18)

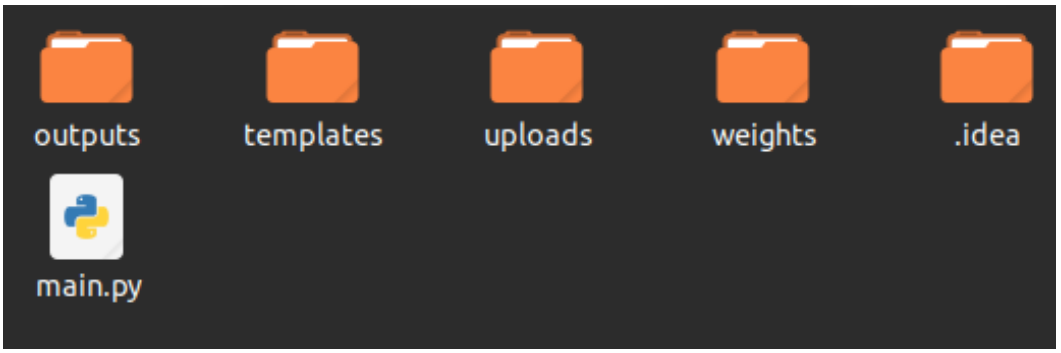


Рисунок 3.18 - Каталог проекту

Після завантаження відео та запуску моделі відео буде оброблятися кадр за кадром. На звичайному процесорі ноутбука один кадр займає в середньому одну секунду, що досить довго, враховуючи швидкість алгоритму YOLO. Однак модель обробляє розмір кадру 4K, що робить обчислювальні ресурси, необхідні для моделі, дуже складними. За

допомогою GPU обробку відео можна прискорити до 30 кадрів в секунду.

Модель тестували через інтерфейс із використанням відкритих кадрів з дрону. (Рисунок 3.19-20).

Рисунок 3.19 - Результати роботи моделі на нових даних

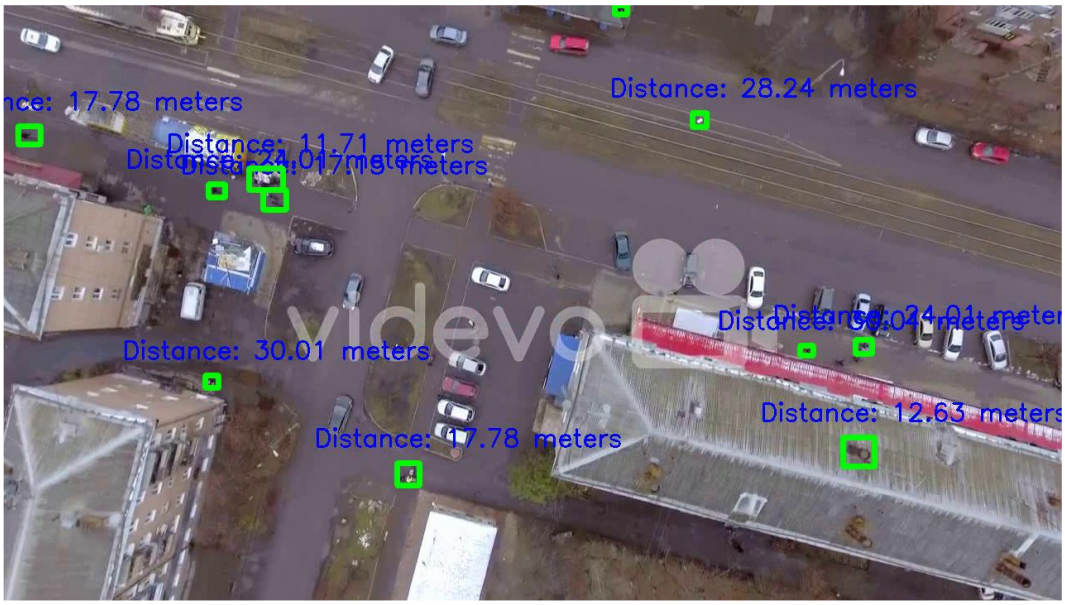


Рисунок 3.20 – Результати моделювання на нових даних

Як ви можете бачити на зображенні вище, модель дуже добре ідентифікує людей, але допускає помилки в класифікації зайвих об’єктів,

таких як димоходи, як людей, що є результатом менш різноманітних даних навчання. Ці помилки можна виправити, перенавчаючи модель на нові типи даних.

У цьому розділі наведено ретельний огляд системи YOLO, яка забезпечує ефективний об’єктний аналіз зображень у реальному часі. За допомогою YOLO була розроблена система визначення відстані до об'єктів (особливо людей) за зображеннями, отриманими з дронів.

У процесі обробки та відбору даних використовуються різні методи для підвищення точності розпізнавання об’єктів. Враховуються об'єкти різного розміру, освітлення та інші аномалії, які можуть вплинути на точність моделі.

Розроблена модель навчається за допомогою мічених даних, що містять зображення людей у різних позах та на різних відстанях від камери. Навчання моделі передбачає ітераційний процес зміни параметрів, вибірки даних і оптимізації алгоритмів для досягнення оптимальної точності. Оцінка моделі показує її великий потенціал. Він показує задовільні результати на зображеннях 4k, отриманих з дронів, і здатний ідентифікувати людей з високою точністю. Однак іноді модель плутає людей з об’єктами, які раніше не спостерігали, наприклад, димоходами. Це може бути пов’язано з обмеженими навчальними даними, які не містять достатньо варіацій для таких об’єктів.

Для подальшого вдосконалення системи рекомендується розширити навчальний набір даних, включивши більше прикладів димоходів та інших невеликих об’єктів, які можна сплутати з людьми.

Результати показують потенціал розробленої моделі та веб-інтерфейсу для взаємодії з нею. Завдяки вдосконаленням алгоритму та додатковим навчальним даним система може широко використовуватися в різних сферах.

ВИСНОВКИ

Бакалаврська робота присвячена проблемі вимірювання відстані до об'єктів у відеоматеріалах. У цьому дослідженні розглядаються різні методи, алгоритми та системи, що використовуються для вимірювання відстаней. Досліджено та проаналізовано поняття відстані, методи відеовиміру відстані, а також існуючі системи та методи для вирішення цієї задачі.

Глава 1 розглядає основні концепції та термінологію, пов’язану з визначенням відстані, і надає огляд існуючих методів вимірювання відстані відео та їхніх характеристик.

Розділ 2 детально вивчає різноманітні аспекти, методи, алгоритми та прийоми, пов’язані з системами вимірювання відстані до об’єктів у відео. Були розглянуті функції бібліотеки OpenCV, вивчена архітектура мережі обробки відео YOLOv8, метод побудови набору даних, метод оптимізації та підвищення точності визначення відстані, а також метод підтримки кількох камер і автоматичного калібрування. вивчав. Крім того, була також досліджена можливість розробки веб-додатків з використанням штучного інтелекту для розпізнавання відеооб’єктів.

У Розділі 3 розроблено систему для вимірювання відстані до об’єктів у відео. Виконав обробку та відбір даних, навчив модель за допомогою алгоритму YOLOv8, оцінив її продуктивність, розробив алгоритм визначення відстані та розробив веб-інтерфейс для взаємодії з моделлю. Результатом цієї роботи стала перспективна модель, яка вміє обробляти зображення 4K з дронів і вміє розпізнавати людей. Однак ми виявили проблему - іноді модель плутала людей об'єктами, наприклад димоходами, які не спостерігалися під час попереднього навчання. Це може бути пов’язано з обмеженими навчальними даними.

Загалом ця стаття дає змогу вивчити та проаналізувати різні аспекти

визначення відстані до об’єктів у відео. Розроблена система та модель показали свій потенціал, але потребують подальшого вдосконалення. Результати досліджень можуть бути використані в різних сферах, таких як системи спостереження, безпеки, військові сфери тощо, де важливо точно визначити відстань до об’єкта або його координати.

Відповідно до реєстраційних стандартів і правил, а також відповідно до різноманітних вимог, завдання бакалаврської роботи було успішно виконано. Проект виконав свою роботу та не мав помилок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Distance. From Wikipedia, the free encyclopedia. Режим доступу: https://en.wikipedia.org/wiki/Distance (Дата звернення 06.05.2024)

2. Elena Serea, Mihai Penciuc, Marinel Costel Temneanu and Codrin Donciu. Video Distance Measurement Technique Using Least Squares Based Sharpness Cost Function. ежим доступу: https://www.mdpi.com/2227-7390/10/18/3273 (Дата звернення 06.05.2024)

3. Harasim Eugen, Codrin Donciu. Video distance measurement based on focus. 2014. Режим доступу: https://ieeexplore.ieee.org/document/9305593 (Дата звернення 06.05.2024)

4. Wei Zhou, Youfeng Zhang, Unhong Zheng, Pingfan Li. Research on distance measurement method of moving objects in video. Режим доступу: https://www.researchgate.net/publication/340312912_Research_on_distance_measurement_method_of_moving_objects_in_video (Дата звернення 06.05.2024)

5. Abdelmoghit Zaarane, Ibtissam Slimani, Wahban Al Okaishi, Issam Atouf, Abdellatif Hamdoun. Distance measurement system for autonomous vehicles using stereo camera. March 2020. Режим доступу: https://www.researchgate.net/publication/340312912_Research_on_distance_measurement_method_of_moving_objects_in_video (Дата звернення 06.05.2024)

6. Ciprian Dragne, Isabela Todirite, Marius Pandelea, Mihaiela Iliescu. Distance Assessment by Object Detection-For Visually Impaired Assistive Mechatronic System. June 2022. Режим доступу: https://www.researchgate.net/publication/361463112_Distance_Assessment_by_Object_Detection- For_Visually_Impaired_Assistive_Mechatronic_System (Дата звернення 06.05.2024)

7. Doha 2010 - Introducing Video Distance Measurement. World Athletics.

Режим доступу: https://worldathletics.org/news/news/doha-2010- introducing-video-distance-measur (Дата звернення 06.05.2024)

8. Computer Vision: Determining the Distance From an Object in a Video. Baeldung. March 29, 2024. Режим доступу: https://www.baeldung.com/cs/cv-compute-distance-from-object-video (Дата звернення 06.05.2024)

9. Gaudenz Boesch. What is OpenCV? The Complete Guide (2024). Режим доступу: https://viso.ai/computer-vision/opencv/ (Дата звернення 06.05.2024)

10. OpenCV: Introduction. OpenCV Documentation. Режим доступу: https://docs.opencv.org/4.x/d1/dfb/intro.html (Дата звернення 06.05.2024)

11. Akshit Mehra. Understanding YOLOv8 Architecture, Applications & Features. Apr 16, 2024. Режим доступу: https://www.labellerr.com/blog/understanding-yolov8-architecture-applications-features/ (Дата звернення 06.05.2024)

12. YOLOv8 for Object Detection Explained [Practical Example]. Encord. Mar 22, 2024. Режим доступу: https://medium.com/cord-tech/yolov8-for-object- detection-explained-practical-example-23920f77f66a (Дата звернення 06.05.2024)

13. Introducing Ultralytics YOLOv8. Ultralytics. Режим доступу: https://docs.ultralytics.com/ (Дата звернення 06.05.2024)

14. Piotr Skalski. How to Train YOLOv8 Object Detection on a Custom Dataset. JAN 10, 2024. Режим доступу: https://blog.roboflow.com/how-to-train- yolov8-on-a-custom-dataset/ (Дата звернення 06.05.2024)

15. Sovit Rath. Train YOLOv8 on Custom Dataset – A Complete Tutorial. Jan 31, 2024. Режим доступу: https://learnopencv.com/train-yolov8-on-custom-dataset/ (Дата звернення 06.05.2024)

16. Train - Ultralytics YOLOv8. Ultralitycs. Режим доступу: https://docs.ultralytics.com/modes/train/ (Дата звернення 06.05.2024)

17. Haidi Zhu, Haoran Wei, Baoqing Li, Xiaobing Yuan and Nasser

Kehtarnavaz. A Review of Video Object Detection: Datasets, Metrics and Methods. 4 November 2020. Режим доступу: https://www.mdpi.com/2076-3417/10/21/7834 (Дата звернення 06.05.2024)

18. Jaskirat Kaur & Williamjeet Singh. Tools, techniques, datasets and application areas for object detection in an image: a review. 23 April 2022. Режим доступу: https://link.springer.com/article/10.1007/s11042-022- 13153-y (Дата звернення 06.05.2024)

19. Top Object Recognition Video Datasets of 2022. Twine AI. Режим доступу: https://www.twine.net/blog/top-object-recognition-video-datasets/ (Дата звернення 06.05.2024)

20. Gousia Habib, Shaima Quresh. Optimization and acceleration of convolutional neural networks: A survey. July 2022. Режим доступу: https://www.sciencedirect.com/science/article/pii/S1319157820304845 (Дата звернення 06.05.2024)

21. J.C. Rodríguez-Quiñonez, O. Sergiyenko, W. Flores-Fuentes, M. Rivas- lopez, D. Hernandez-Balbuena, R. Rascón, P. Mercorelli. Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. May 2017. Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/S1230340217300100 (Дата звернення 06.05.2024)

22. Yong Ye, Yuting Liu, Weihan Yin, Jiahao Deng & Xiaofeng Zhu. Improving the measurement accuracy of distance and positioning for capacitive proximity detection in human-robot interaction. 19 April 2021. Режим доступу: Режим доступу: https://link.springer.com/article/10.1007/s00542-021-05223-2%D1%84 (Дата звернення 06.05.2024)

23. Zewei Liu, Dongming Lu, Weixian Qian, Kan Ren, Guohua Gu, Jun Zhang & Chen Mao. A new method for increasing accuracy of distance measurement based on single visual camera. 21 February 2019. Режим доступу: https://link.springer.com/article/10.1007/s11082-019-1786-z (Дата звернення

‘

06.05.2024)

24. Realtime Distance Estimation Using OpenCV – Python. GeeksForGeeks. Режим доступу: https://www.geeksforgeeks.org/realtime-distance- estimation-using-opencv-python/ (Дата звернення 06.05.2024)

25. Measuring distance between two contours in video? OpenCV Python. StackOverflow. Режим доступу: https://stackoverflow.com/questions/41832237/measuring-distance-between-two-contours-in-video-opencv-python (Дата звернення 06.05.2024)

26. Haseeb Muhammad Abdul, Ristić-Durrant Danijela, Gräser Axel, Banić Milan & Stamenković Dušan. Multi-DisNet: Machine Learning-Based Object Distance Estimation from Multiple Cameras. 23 November 2019. Режим доступу: https://link.springer.com/chapter/10.1007/978-3-030-34995-0_41 (Дата звернення 06.05.2024)

27. Eric J. Howe, Stephen T. Buckland, Marie-Lyne Després-Einspenner, Hjalmar S. Kühl. Distance sampling with camera traps. 10 May 2017. Режим доступу: https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.12790 (Дата звернення 06.05.2024)

28. Amelia Azman, M. Hanafi Ani. Object Distance and Size Measurement Using Stereo Vision System. December 2012. Режим доступу: https://www.researchgate.net/publication/274773924_Object_Distance_and_ Size_Measurement_Using_Stereo_Vision_System (Дата звернення 06.05.2024)

29. Apar Garg. Stereo Vision: Depth Estimation between object and camera. Nov 17, 2021. Режим доступу: https://www.researchgate.net/publication/274773924_Object_Distance_and_ Size_Measurement_Using_Stereo_Vision_System (Дата звернення 06.05.2024)

30. Paul Pias. Object-Detection-and-Distance-Measurement. Github. Режим доступу: https://github.com/paul-pias/Object-Detection-and-Distance-

Measurement (Дата звернення 06.05.2024)

31. NEDA CVIJETIC. To Go the Distance, We Built Systems That Could Better Perceive It

32. See how deep neural networks trained on radar and lidar data predict 3D distances from 2D images. Nvidia DRIVE Labs. Режим доступу: https://blogs.nvidia.com/blog/2019/06/19/drive-labs-distance-to-object-detection/ (Дата звернення 06.05.2024)

33. Hans-Gerd Maas. Image sequence based automatic multi-camera system calibration techniques. December 1999. Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/S0924271699000295 (Дата звернення 07.05.2024)

34. Zhen Liu, Fengjiao Li, Guangjun Zhang. An external parameter calibration method for multiple cameras based on laser rangefinder. January 2014. Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/S0263224113005289 (Дата звернення 07.05.2024)

35. Adrian Rosebrock. Find distance from camera to object/marker using Python and OpenCV. January 19, 2015. Режим доступу: https://pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/ (Дата звернення 07.05.2024)

36. Flask. PyPi. Режим доступу: https://pypi.org/project/Flask/ (Дата звернення 07.05.2024)

37. Nakul Lakhotia. Video Streaming in Web Browsers with OpenCV & Flask. Oct 20, 2020. Режим доступу: https://towardsdatascience.com/video-streaming-in-web-browsers-with-opencv-flask-93a38846fe00 (Дата звернення 07.05.2024)

38. Mark Winterbottom, Brooke Rutherford. Flask vs. Django: Which Web Framework is Best? Режим доступу: https://blog.udemy.com/flask-vs- django/ (Дата звернення 07.05.2024)

‘

39. Daniel Diaz. 25 Python Frameworks to Master in 2024. May 3, 2024. Режим доступу: https://kinsta.com/blog/python-frameworks/ (Дата звернення 07.05.2024)

40. Irfan Alghani Khalid. How to Display Video Streaming From A Webcam Using Flask. Aug 30, 2021. Режим доступу: https://towardsdatascience.com/how-to-display-video-streaming-from-a-webcam-using-flask-7a15e26fbab8 (Дата звернення 07.05.2024)

41. Gerry Christian Ongko. Building a Machine Learning Web Application Using Flask. Feb 3, 2022. Режим доступу: https://towardsdatascience.com/building-a-machine-learning-web- application-using-flask-29fa9ea11dac (Дата звернення 07.05.2024)

42. Diane Phan. What's Cookin'? Build an Image Recognition App on WhatsApp using Twilio MMS, Clarifai API, Python, and Flask 07 oct, 2020. Режим доступу: https://www.twilio.com/blog/image-recognition-whatsapp-mms- python-flask-clarifai (Дата звернення 07.05.2024)

43. Sachin Pal. Build Flask App For Image Recognition Using Deep Learning Model. Sachin Pal. Nov 17, 2022. Режим доступу: https://geekpython.in/flask-app-for-image-recognition (Дата звернення 07.05.2024)

‘

ДОДАТКИ

‘

Додаток А

(обов'язковий)

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: _« Автоматизована система опрацювання сенсорної відеоінформації для контролю відстані між об'єктами»_____

Тип роботи: _____Бакалаврська дипломна робота_____
(БДР, МКР)

Підрозділ _____КСУ, ФІТА_____
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність ___98,6%_____ Схожість__1,4%_____

Аналіз звіту подібності (відмітити потрібне)

☐ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Володимир ДУБОВОЙ___
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichek щодо роботи.

Автор роботи _____ _____Дмитро ШТУРМА_____
(підпис) (прізвище, ініціали)

Керівник роботи _____ _____Олена НИКИТЕНКО_____
(підпис) (прізвище, ініціали)

Додаток А

(обов'язковий)

ВНТУ

ЗАТВЕРДЖЕНО
Зав. кафедри КСУ ВНТУ,
д.т.н., проф.

_____В'ячеслав КОВТУН

" 24 " травня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської дипломної роботи

АВТОМАТИЗОВАНА СИСТЕМА ОПРАЦЮВАННЯ СЕНСОРНОЇ ВІДЕОІНФОРМАЦІЇ ДЛЯ КОНТРОЛЮ ВІДСТАНІ МІЖ ОБ'ЄКТАМИ

Студент групи

_____АКІТ-22мс Дмитро ШТУРМА

Керівник

_____д.т.н., доц. Олена НИКИТЕНКО

Вінниця 2024

1. Назва та галузь застосування

1.1. Назва – Автоматизована система опрацювання сенсорної відеоінформації для контролю відстані між об’єктами

1.2. Галузь застосування – Навчальні процеси.

2. Підстава для проведення розробки.

Тема бакалаврської дипломної роботи затверджена наказом по ВНТУ №_80_ від _11 березня_.2024 р.

3. Мета та призначення розробки.

Метою цієї роботи є не лише опис рішень для відеовиміру відстані, але й дослідження їх ефективності та потенціалу застосування

4. Джерела розробки.

Бакалаврська дипломна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

Richard Szeliski. Computer Vision: Algorithms and Applications. – Springer, 2010. – С. 99-300. [https://www.cs.ccu.edu.tw/~damon/tmp/SzeliskiBook_20100903_draft.pdf].

Guide to Object Detection. 2020. [Електронний ресурс] – Режим доступу до ресурсу: [https://towardsdatascience.com/going-deep-into-object-detection-bed442d92b34]

Ranjay Krishna Computer Vision: Foundations and Applications. 2012. – С. - 300. [http://vision.stanford.edu/teaching/cs131_fall1718/files/cs131-class-notes.pdf]

LeCun, Y., Bengio, Y., & Hinton, G. Deep learning. Nature, 521(7553), – 2015. – С. 436-444.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

– Моделювання
– Пошук локацій
– Обробка зображень

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- WINDOWS 10;
- Потужна відеокарта

5.2.2. Умови експлуатації системи:

- робота на мобільних додатках;
- можливість цілодобового функціонування системи;

‘

- дані оновлюються і є актуальними.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об’єкті управління відповідно до теми дипломної роботи. Постановка задач дослідження «15» травня 204 р.
2. Визначення технічних характеристик системи «19» травня 2024 р.
3. Розробка програмного забезпечення системи «30» травн 2024 р.

6.2 Графічні матеріали:

1. Розробка моделі системи «24» травня 2024 р.
2. Тестування програмного забезпечення «10» червня 2024 р.

7. Порядок контролю і приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «4» червня 2024 р.

7.2. Атестація проєкту здійснюється на попередньому захисті. Попередній захист бакалаврської дипломної роботи провести до «10» червня 2024 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист бакалаврської дипломної роботи провести до «20»червня 2024 р.

Додаток В Лістинг програми

Лістиги

```
from flask import Flask, render_template, request, send_file
from werkzeug.utils import secure_filename
import os
from ultralytics import YOLO
import cv2
import shutil
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'uploads'
app.config['OUTPUT_FOLDER'] = 'outputs'
app.config['ALLOWED_EXTENSIONS'] = {'mp4', 'pt'} # Specify allowed file
extensions
yolo_weights = '' # Global variable to store the selected YOLO weights file
path
def allowed_file(filename):
return '.' in filename and filename.rsplit('.', 1)[1].lower() in app.config['ALLOWED_EXTENSIONS']
@app.route('/')
def index():
return render_template('index.html')
@app.route('/upload', methods=['POST'])
def upload():
global yolo_weights
file = request.files['video']
yolo_weights_file = request.form['weights']
if file and allowed_file(file.filename) and yolo_weights_file:
filename = secure_filename(file.filename)
file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
yolo_weights = yolo_weights_file
print(filename, yolo_weights)
filename = process_video(filename)
return render_template('download.html',
```

‘

```
filename=os.path.basename(filename)) else:
return "Invalid file format. Please upload an MP4 video and choose a YOLO weights file."
def process_video(video_path):
model = YOLO(f'weights/{yolo_weights}')
print(video_path)
video = cv2.VideoCapture('uploads/' + video_path)
frame_width = int(video.get(cv2.CAP_PROP_FRAME_WIDTH)) frame_height = int(video.get(cv2.CAP_PROP_FRAME_HEIGHT)) fps = video.get(cv2.CAP_PROP_FPS)
output_path = f'outputs/processed_{video_path}'
output_video = cv2.VideoWriter(output_path, cv2.VideoWriter_fourcc(*'mp4v'), fps, (frame_width, frame_height))
while True:
ret, frame = video.read()
if not ret: break
print('read')
results = model(frame)[0]
print('framed')
for box in results.boxes.xyxy:
x1, y1, x2, y2 = map(int, box)
# Calculate the distance to the object
focal_length = 1200.29 # Focal length of the camera (in mm)
object_width = 0.4 # Width of the object (in meters)
distance = (focal_length * object_width) / (x2 - x1)
# Draw the rectangle
cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 5)
cv2.putText(frame, f"Distance: {distance:.2f} meters", (x1-100, y1-20), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
output_video.write(frame)
if os.path.exists(output_path):
return output_path
else:
print(video_path, output_path)
shutil.copy2('uploads/' + video_path, output_path)
return output_path
@app.route('/download/<filename>')
def download(filename):
file_path = os.path.join(app.config['OUTPUT_FOLDER'], filename)
if os.path.exists(file_path):
```

‘

```
return send_file(file_path, as_attachment=True) else:
return "File not found."
if __name__ == '__main_
app.run(debug=True)
```

‘

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

«АВТОМАТИЗОВАНА СИСТЕМА ОПРАЦЮВАННЯ СЕНСОРНОЇ ВІДЕОІНФОРМАЦІЇ ДЛЯ КОНТРОЛЮ ВІДСТАНІ МІЖ ОБ'ЄКТАМИ»»

Виконав: студент 4-го курсу,
групи АКІТ-22мс
спеціальності 151 –
Автоматизація такомп'ютерно-інтегровані технології
(шифр і назва спеціальності)

_____Дмитро ШТУРМА
(ім'я та прізвище)

Керівник: к.т.н., доцент каф. КСУ
_____Олена Никитенко
(ім'я та прізвище)
«10» 06 2024р.

Рецензент: к.т.н., доц. каф. АІІТ
_____Юрій ІВАНОВ
(ім'я та прізвище)
«11» червня 2024 р.

Вінниця - 2024

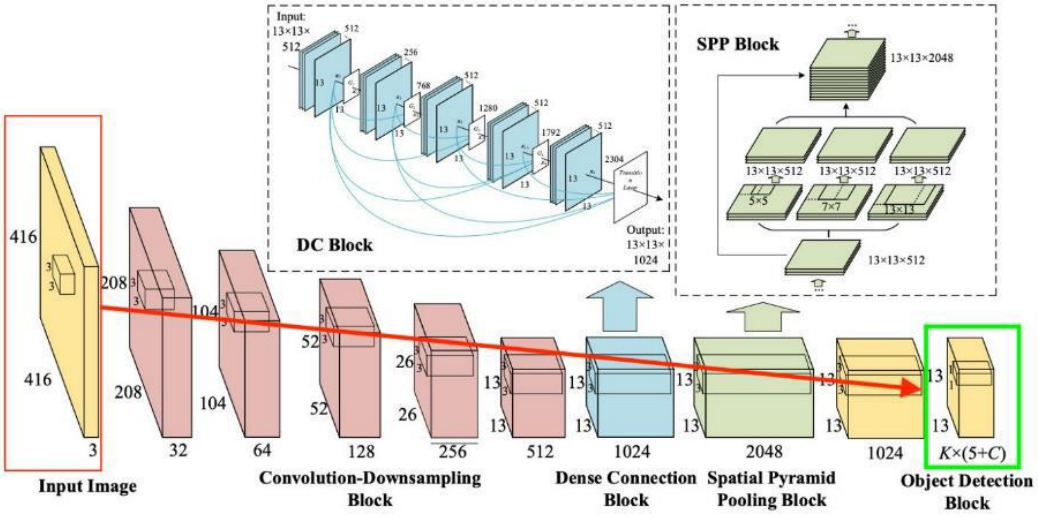
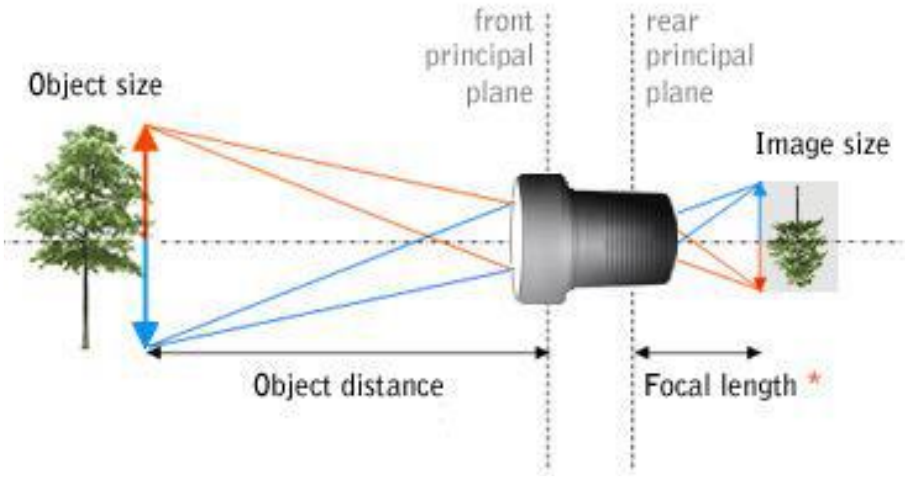
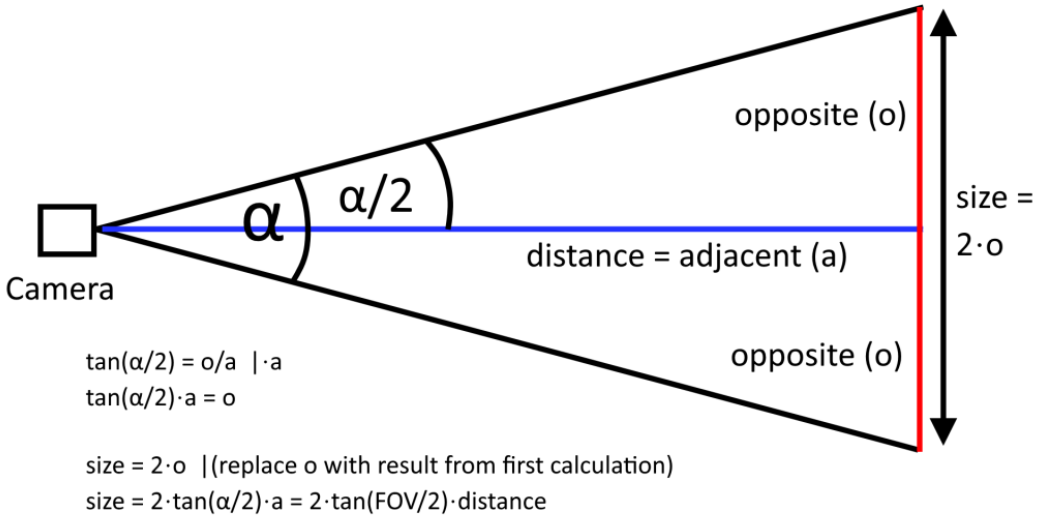


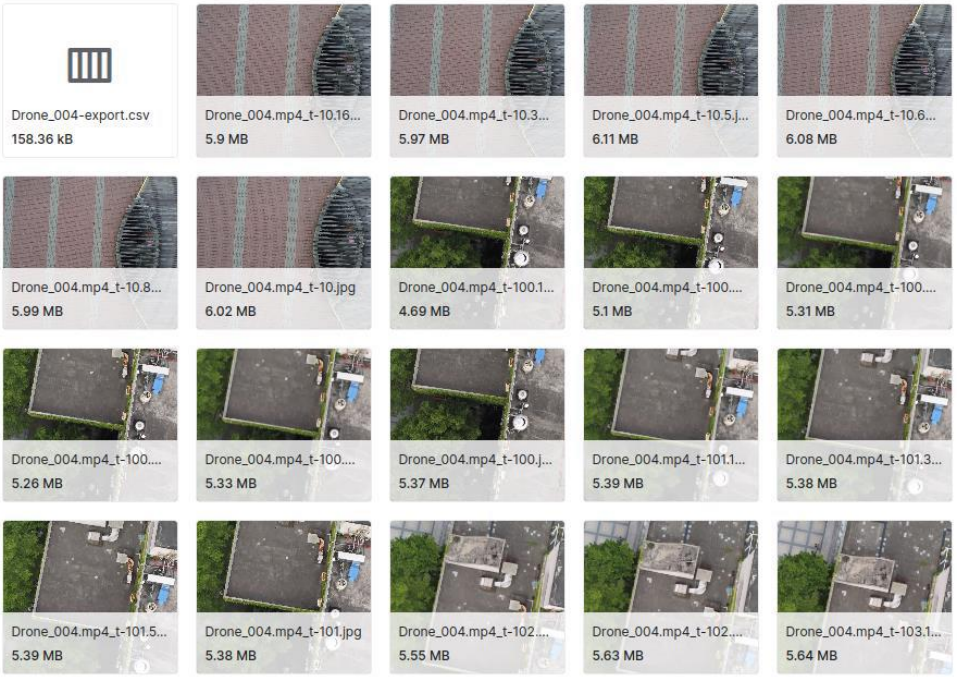
Схема обробки зображення в YOLOv8



Демонстрація алгоритму на основі фокусної відстані [35]

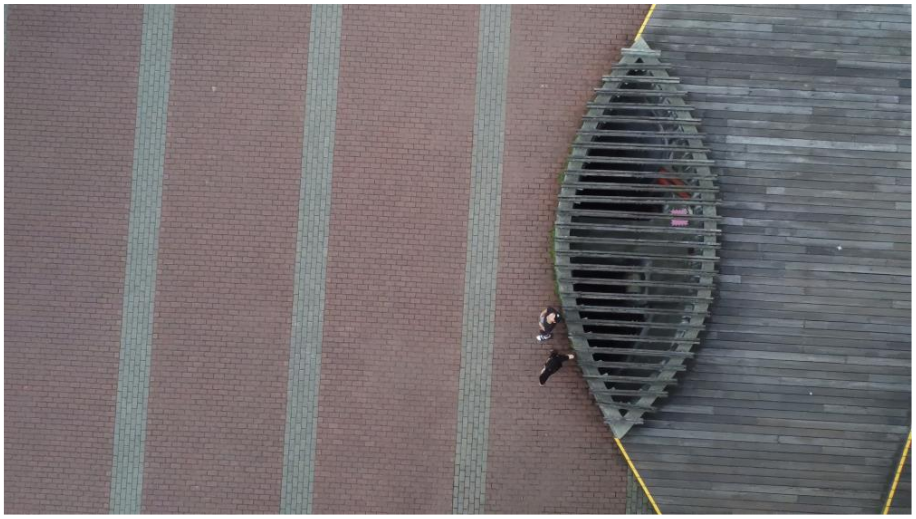


Демонстрація алгоритму вимірювання відстані на основі пікселів



Перегляд зображень у наборі даних

‘



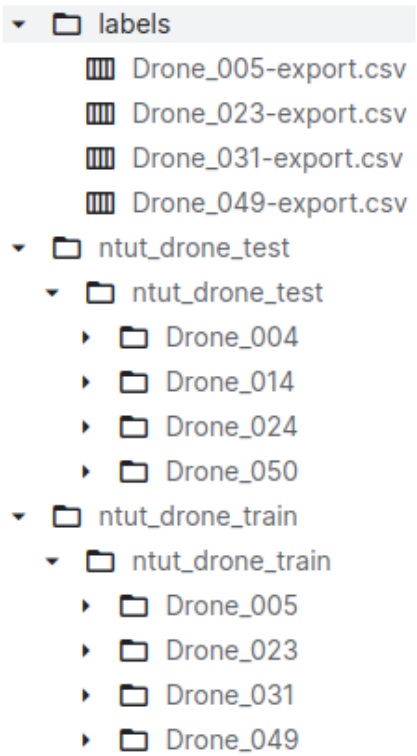
Одне із зображень, наданих у наборі даних.

Вид зображення з мітками

‘

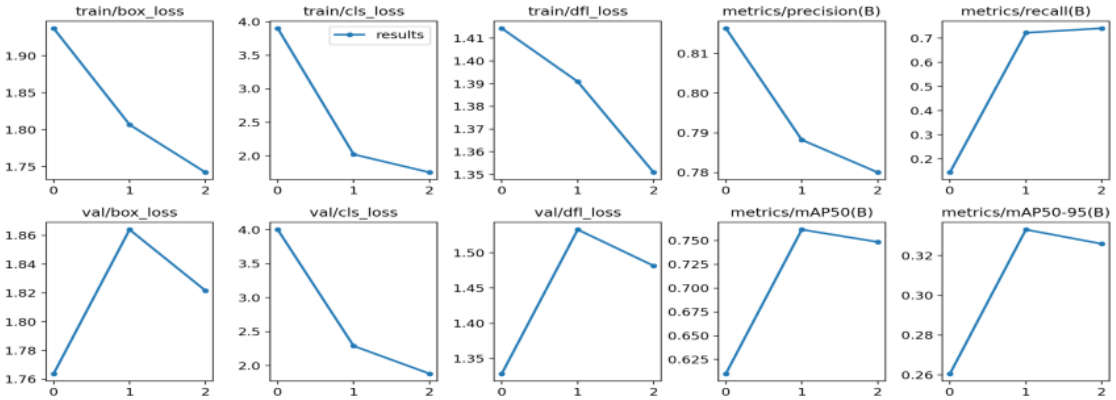
image	xmin	ymin	xmax	ymax	label
Drone_005.mp4_t-12.833333	2195.703128246 2035	159.44943494639 233	2343.904368769 9805	299.9917143581 57	walk
Drone_005.mp4_t-12.833333	2195.703128246 2035	159.44943494639 233	2343.904368769 9805	299.9917143581 57	push
Drone_005.mp4_t-12.833333	2485.488565488 5675	220.78817733990 147	2626.528066528 0675	367.0935960591 133	walk
Drone_005.mp4_t-12.666667	2219.653232196 3073	154.12923790205 735	2367.854472720 0842	294.6715173138 2205	walk
Drone_005.mp4_t-12.666667	2219.653232196 3073	154.12923790205 735	2367.854472720 0842	294.6715173138 2205	push
Drone_005.mp4_t-12.666667	2509.438669438 6712	207.48768472906 403	2650.478170478 1713	353.7931034482 758	walk
Drone_005.mp4_t-12.5	2243.603336146 411	135.50854824688 494	2391.804576670 188	276.0508276586 496	walk
Drone_005.mp4_t-12.5	2243.603336146 411	135.50854824688 494	2391.804576670 188	276.0508276586 496	push
Drone_005.mp4_t-12.5	2544.033264033 2654	186.20689655172 413	2685.072765072 7654	332.5123152709 359	walk
Drone_005.mp4_t-12.333333	2259.570072113 147	114.22776006954 504	2407.771312636 9235	254.7700394813 0974	walk
Drone_005.mp4_t-12.333333	2259.570072113 147	114.22776006954 504	2407.771312636 9235	254.7700394813 0974	push

Перегляд файлу даних зображення

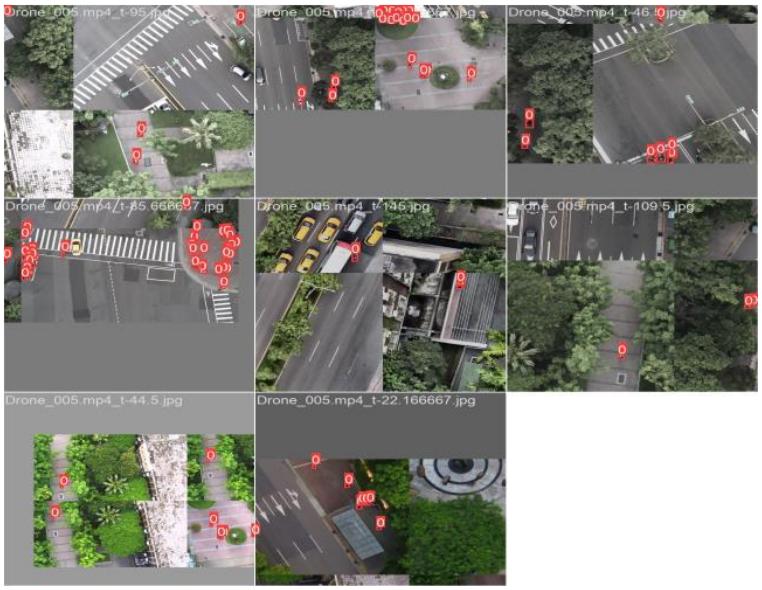


Структура каталогу набору даних

‘

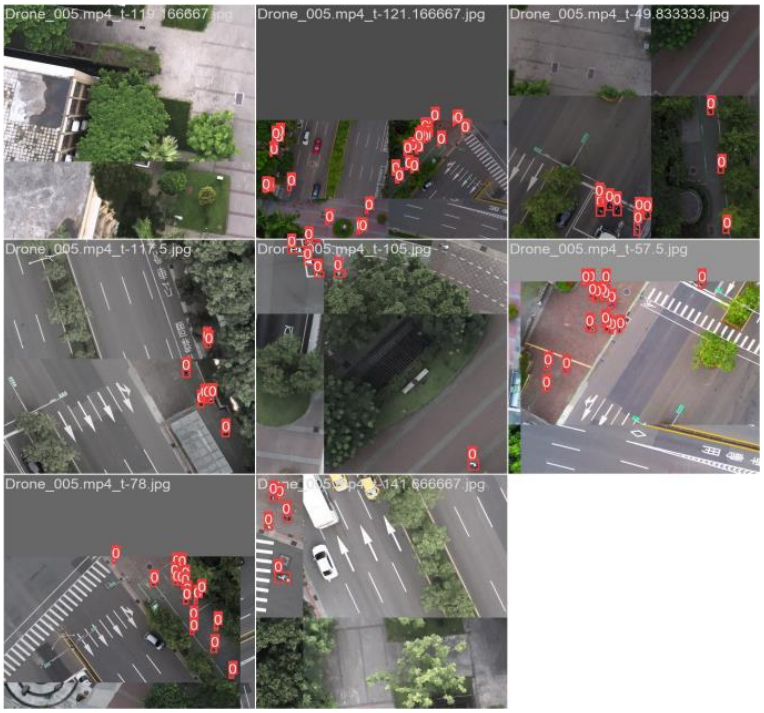


Візуалізація таблиці



Результати тренування Batch1

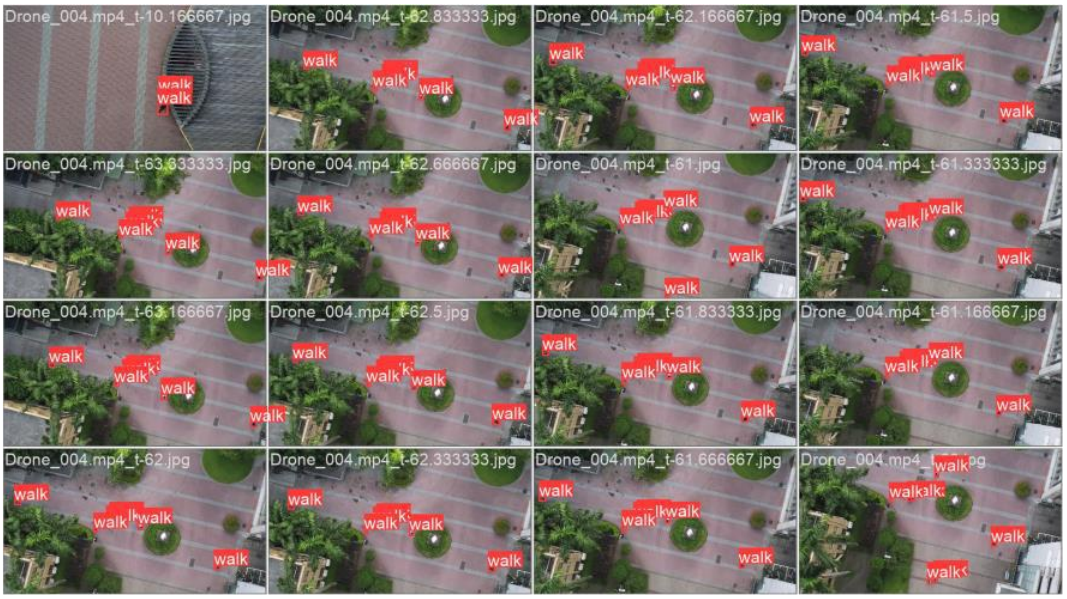
‘



Результати тренування Batch2

Передбачені мітки даних на val_batch2

‘



Правдиві мітки даних на val_batch1

‘