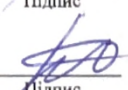
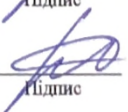


КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод визначення швидкості руху автомобіля у відеопотоці
нейромережевими засобами

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент групи КН-21-1  Микола МЕЛЬНИК
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ
Керівник: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ
Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

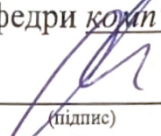
зав. кафедри КН, д.т.н., професор

19 06 2025 р.

 Олександр БАРМАК
Підпис Ім'я, ПРІЗВИЩЕ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук


(підпис)
д.т.н., професор Олександр БАРМАК
«10» 02 2025 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Метод визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами»

2. Завдання видано студенту Миколі МЕЛЬНИКУ
(Ім'я, прізвище)

3. Керівник роботи к.т.н., доц. каф. КН Руслан БАГРІЙ
(посада, ім'я, прізвище)

4. Затверджено наказом університету від «07» 02 2025 р. № 23

5. Дата видачі завдання студенту: «10» 02 2025 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета кваліфікаційної роботи бакалавра – підвищення точності визначення швидкості руху автомобіля у відеопотоці засобами глибокого навчання. Для досягнення поставленої мети необхідно розв'язати такі завдання: провести аналіз існуючих методів комп'ютерного зору та глибокого навчання, що застосовуються для виявлення й трекінгу транспортних засобів у відеопотоці; розробити метод визначення швидкості руху виявлених автомобілів на основі координат об'єктів і часових міток кадрів; підібрати та підготувати відомий набір даних із зображень автомобілів, що включає різні умови зйомки; створити програмну реалізацію методу з підтримкою детекції, трекінгу, обчислення швидкості та візуалізації результатів; провести експериментальні дослідження для оцінки точності запропонованого методу.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2025	Виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	лютий 2025	Виконано
3	Проектування та розробка загальної архітектури програмного забезпечення, інтерфейсу користувача, вибір засобів реалізації програмного забезпечення	березень 2025	Виконано
4	Створення та тестування програмного забезпечення	квітень 2025	Виконано
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення згідно вимог	травень 2025	Виконано
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2025	Виконано
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2025	Виконано
8	Підготовка до захисту та захист кваліфікаційної роботи бакалавра	червень 2025	Виконано

Виконавець: студент групи КН-21-1

Група виконавця

Підпис

Микола МЕЛЬНИК

Ім'я, ПРІЗВИЩЕ

Керівник:

к.т.н., доц. каф. КН

Науковий ступінь, посада

Підпис

Руслан БАГРІЙ

Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-21-1
Микола МЕЛЬНИК

Керівник кваліфікаційної роботи бакалавра: к.т.н., доц. каф. КН Руслан
Багрій

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
55	27	2	42	2

Мета кваліфікаційної роботи бакалавра – підвищення точності визначення швидкості руху автомобіля у відеопотоці шляхом застосування нейромережових засобів комп'ютерного зору. Розроблено метод автоматичного визначення швидкості руху транспортних засобів, що ґрунтується на послідовному виявленні об'єктів, їхньому трекінгу, аналізі зміщення та обчисленні швидкості з урахуванням частоти кадрів і масштабного коефіцієнта (PPM).

Метод реалізовано у вигляді застосунку на Python із графічним інтерфейсом Tkinter. Для детекції об'єктів використано модель YOLOv10, для трекінгу – алгоритм DeepSORT, для обробки відео – бібліотеку OpenCV.

Результатом є програмне рішення, що дає змогу визначати швидкість автомобіля на відео без використання зовнішніх сенсорів. Застосунок виконує роль тестового середовища для оцінки ефективності розробленого методу.

Ключові слова: відеопотік, швидкість автомобіля, трекінг, YOLOv10, DeepSORT, OpenCV, Python, Tkinter, комп'ютерний зір, глибоке навчання.

Виконавець: студент групи КН-21-1

Група виконавця


Підпис

Микола МЕЛЬНИК

Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень.....	3
Вступ.....	4
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій .6	6
1.1 Аналіз інформаційних моделей.....	6
1.2 Огляд теоретичних підходів до визначення швидкості руху за відеопотоками ..7	7
1.3 Аналіз існуючих програмних засобів для виявлення швидкості транспортних засобів	11
1.4 Мета та завдання кваліфікаційної роботи.....	14
Розділ 2 Проектування методу визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами	15
2.1 Загальна схема методу визначення швидкості руху автомобіля у відеопотоці .15	15
2.2 Архітектура нейромережевої моделі для обробки відеопотоку	18
2.3 Підготовка даних для донавчання моделі.....	21
2.4 Виявлення об'єктів у відеопотоці	24
2.5 Визначення швидкості руху автомобілів	26
2.6 Метрики оцінювання якості моделі	31
Висновки до розділу 2	34
Розділ 3 Програмна реалізація методу визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами.....	35
3.1 Засоби розробки програмного застосунку.....	35
3.2 Структура та особливості реалізації програмного застосунку	36
3.3 Аналіз результатів детекції об'єктів класу «автомобіль»	39
3.3.1 Просторовий та розмірний розподіл міток у навчальній вибірці.....	39
3.3.2 Результати тестування моделі YOLO.....	41
3.4 Оцінка точності визначення швидкості транспортних засобів	48
Висновки до розділу 3	50
Загальні висновки	51
Перелік посилань	52
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
CNN	Convolutional Neural Network – згорткова нейронна мережа.
YOLO	You Only Look Once
MAE	Mean Absolute Error – середня абсолютна похибка.
PPM	Pixels Per Meter – пікселів на метр.
RMSE	Root Mean Square Error – корінь із середньоквадратичної похибки.
UI	User Interface – інтерфейс користувача.
FPS	Frames Per Second – кількість кадрів за секунду.

Вступ

Кваліфікаційна робота присвячена розробці методу визначення швидкості руху автомобіля у відеопотоці. У сучасних умовах інтенсивного зростання автомобільного трафіку в містах особливої актуальності набувають ефективні, недорогі та масштабовані рішення для контролю швидкісного режиму. Безпечна організація дорожнього руху вимагає постійного моніторингу швидкості транспортних засобів, однак традиційні технічні засоби, такі як радары, лазерні далекоміри чи GPS-навігація, є затратними, мають обмежене покриття та потребують встановлення додаткового обладнання.

Водночас системи відеоспостереження вже давно стали невід'ємною частиною міської інфраструктури, забезпечуючи візуальний контроль за дорожнім рухом на перехрестях, автомагістралях, у зонах підвищеної небезпеки та біля соціально важливих об'єктів. Камери встановлюються для фіксації порушень, спостереження за трафіком, реагування на надзвичайні ситуації, а також для збору даних з метою аналітики. Велика кількість вже встановленого обладнання створює унікальні можливості для повторного використання отриманих відеозаписів не лише з моніторинговою, але й з аналітичною метою.

Сучасні досягнення у сфері комп'ютерного зору, зокрема глибокі згорткові нейронні мережі, відкривають широкі можливості для розробки таких систем. Однією з найуспішніших архітектур для задач об'єктного виявлення є YOLO (You Only Look Once), яка дозволяє у реальному часі визначати розташування транспортних засобів у кадрі та класифікувати їх. У поєднанні з алгоритмами трекінгу, такими як DeepSORT, що забезпечують збереження ідентичності об'єкта між кадрами, можливо формувати траєкторію руху окремого автомобіля. Аналіз координат зміщення та часових міток дає змогу оцінити пройденої відстань і швидкість руху кожного транспортного засобу з високою точністю – виключно на основі відео, без використання фізичних сенсорів, таких як радары чи GPS.

У межах цієї кваліфікаційної роботи передбачається дослідити та розробити метод визначення швидкості руху автомобіля у відеопотоці.

Планується здійснити масштабування піксельних зміщень у фізичні одиниці за допомогою коефіцієнта PPM, що визначається на основі геометрії сцени (висоти камери, кута огляду тощо), а також використати частоту відео для врахування часових інтервалів.

Підсумком стане створення програмного прототипу у вигляді застосунку, який забезпечуватиме базову функціональність: детекцію, трекінг та обчислення швидкості руху об'єктів у відеопотоці з подальшою візуалізацією результатів. Такий підхід має потенціал для практичного застосування в системах автоматизованого контролю швидкості на базі звичайних камер відеоспостереження.

Об'єкт дослідження – процес визначення швидкості руху автомобіля у відеопотоці.

Предмет дослідження – методи глибокого навчання та комп'ютерного зору для визначення швидкості руху автомобіля у відеопотоці.

Мета кваліфікаційної роботи бакалавра – підвищення точності визначення швидкості руху автомобіля у відеопотоці засобами глибокого навчання.

Завдання кваліфікаційної роботи:

- 1) провести аналіз існуючих методів комп'ютерного зору та глибокого навчання для виявлення та трекінгу транспортних засобів у відеопотоці;
- 2) розробити метод визначення швидкості руху виявлених автомобілів у відеопотоці на основі координат і часових міток кадрів;
- 3) підібрати та підготувати відомий набір даних із зображень автомобілів, що включає різні умови зйомки;
- 4) створити програму реалізацію методу з підтримкою детекції, трекінгу, розрахунку швидкості та візуалізації результатів;
- 5) провести експерименти для оцінки точності запропонованого методу визначення швидкості руху автомобіля.

Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій

1.1 Аналіз інформаційних моделей

Предметна область розпізнавання швидкості автомобіля у відеопотоці охоплює процеси спостереження, виявлення та аналізу руху транспортних засобів з метою визначення їхньої швидкості. Такий підхід є актуальним для систем відеоспостереження, транспортної аналітики та безпеки дорожнього руху, оскільки пропонує доступну та неінвазивну альтернативу традиційним радарним і лазерним засобам фіксації. Відеоаналітичні системи, що базуються на алгоритмах комп'ютерного зору і глибокого навчання, дозволяють здійснювати автоматичний контроль за допомогою звичайних камер, не потребуючи складної інфраструктури.

Ці технології активно застосовуються в «розумних містах», на дорогах, вулицях, паркінгах і контрольованих зонах. У центрі інформаційної моделі перебуває транспортний засіб, для якого фіксуються координати, траєкторія та швидкість на основі аналізу послідовних кадрів відео з часовими мітками. Розрахунок швидкості здійснюється шляхом аналізу зміни положення об'єкта в часі, що особливо ефективно при використанні області інтересу – сегменту кадру, де ведеться детекція та трекінг [1].

Подібні системи істотно полегшують роботу операторів відеоспостереження, автоматизуючи процес відстеження транспортних засобів у потоці відеоданих. Завдяки використанню комп'ютерного зору та алгоритмів машинного навчання, такі рішення здатні самостійно визначати швидкість руху, виявляти порушення правил дорожнього руху та формувати повідомлення про інциденти в режимі реального часу. Це звільняє операторів від рутинної та виснажливої роботи з постійного моніторингу відео і дозволяє їм зосередитися на оперативному аналізі та прийнятті рішень у критичних ситуаціях, тим самим підвищуючи ефективність системи безпеки на дорогах.

1.2 Огляд теоретичних підходів до визначення швидкості руху за відеопотоками

Одним із основних напрямів у галузі автоматизованого аналізу відео та контролю транспортного потоку є використання згорткових нейронних мереж (CNN) для виявлення транспортних засобів у кадрах відеопотоку є використання згорткових нейронних мереж (CNN) для виявлення транспортних засобів на відео та алгоритмів трекінгу, що дозволяють відстежувати їх переміщення між кадрами[4]. На основі зміни координат об'єкта у часі та відомої частоти кадрів можна обчислити швидкість руху транспортного засобу [4].

YOLO (You Only Look Once) – один із найпоширеніших алгоритмів для детекції об'єктів у реальному часі [2]. Архітектура дозволяє обробляти зображення за один прохід, миттєво видаючи координати та клас об'єкта. на рисунку 1.1 буде зображено схему роботи моделі YOLO на кадрі відео.

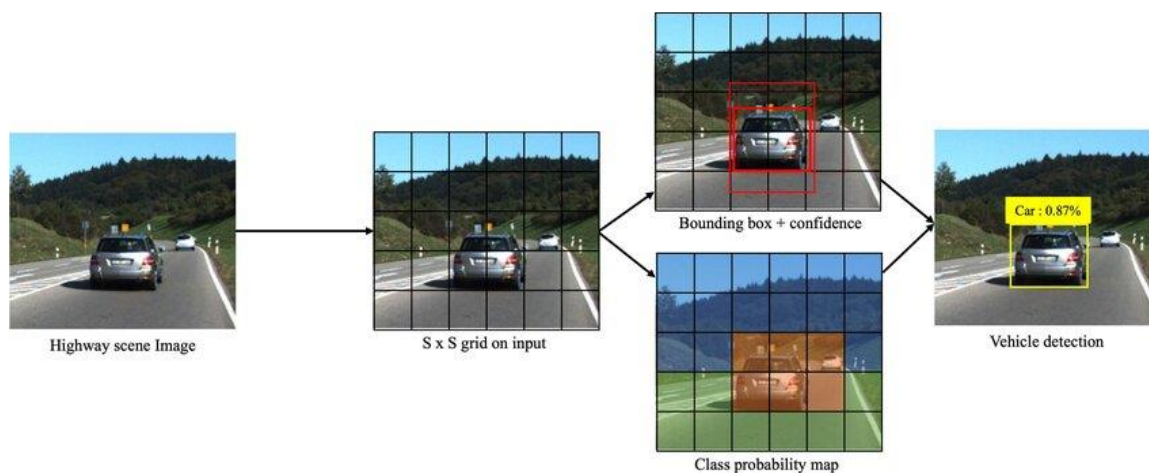


Рисунок 1.1 – Схема роботи моделі YOLO на кадрі відео [41]

Кожне зображення розбивається на сітку, у межах якої детектор визначає наявність транспортних засобів та їх положення.

DeepSORT є розширенням стандартного алгоритму SORT, доповненого глибокою нейронною мережею для кращої асоціації об'єктів між кадрами [3]. Це дозволяє уникнути повторної ідентифікації транспортного засобу, навіть якщо він тимчасово зникає з кадру, на рисунку 1.2 зображено принцип роботи DeepSORT.

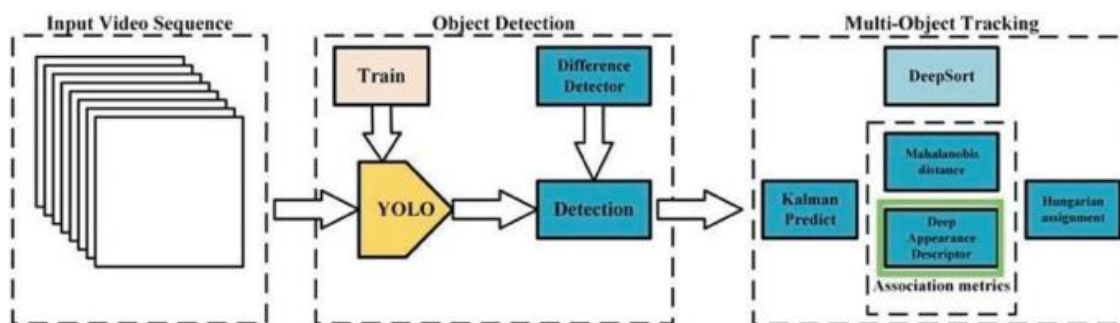


Рисунок 1.2 – Принцип роботи DeepSORT [42]

На рисунку 1.2 зображено структурну схему алгоритму DeepSORT, що поєднує детекцію об'єктів за допомогою нейронної мережі YOLO із системою багатооб'єктного трекінгу. Потік обробки відео починається з детектора, який виявляє об'єкти на кожному кадрі. Далі координати об'єктів передаються до трекера DeepSORT, що використовує Kalman-фільтр для прогнозування наступного положення, а також візуальні дескриптори для оцінки подібності. Модуль Hungarian Assignment відповідає за призначення ідентифікаторів (ID) об'єктам на основі поєднання просторової близькості та візуальної схожості, що дозволяє відслідковувати об'єкти в часі. Такий підхід забезпечує високу стійкість трекінгу навіть за наявності тимчасових втрат об'єктів чи їх часткового перекриття. DeepSORT дозволяє ефективно формувати траєкторії руху, що є критично важливим для задач визначення швидкості транспортних засобів. На рисунку 1.3 показано приклад альтернативного підходу до обробки відео.



Рисунок 1.3 – Обробка відео Optical Flow [5]

Оптичний потік (Optical Flow) – один із класичних методів оцінки руху пікселів між двома сусідніми кадрами [5]. Він дозволяє оцінити, як змінюється положення окремих точок зображення з часом, що є особливо корисним у випадках, коли об'єкт тимчасово недоступний для точного трекінгу або погано детектується. Наприклад, метод Лукаса-Канаде або Farnebäck активно використовується для аналізу локального руху. На практиці його застосовують як допоміжний інструмент для уточнення траєкторії або перевірки стабільності трекінгу, особливо при складних умовах зйомки (наприклад, слабе освітлення чи часткове перекриття об'єктів).

GoodFeaturesToTrack (GFTT) – це алгоритм, який дозволяє визначити ключові точки (corners) на об'єкті, що добре піддаються трекінгу [6]. У задачі визначення швидкості GFTT може використовуватись для виділення точок на кузові або колесах транспортного засобу, які потім супроводжуються у часі для обчислення швидкості. Цей метод є ефективним при попередньому аналізі відео або в поєднанні з трекерами на основі оптичного потоку, на рисунку 1.4 показано визначення точок алгоритму.

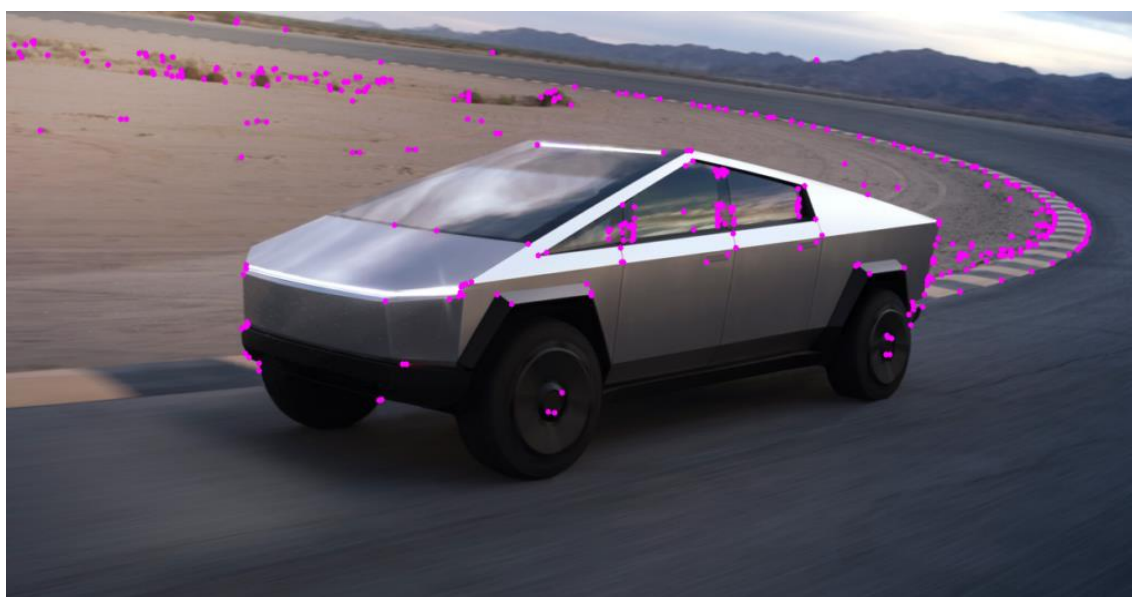


Рисунок 1.4 – GoodFeaturesToTrack (GFTT) [6]

Алгоритм GoodFeaturesToTrack (GFTT) був запропонований Ши-Томасом як ефективний підхід для виявлення добре відслідковуваних точок у зображенні [6]. Його суть полягає у знаходженні таких пікселів, які мають високий градієнт інтенсивності в обох напрямках – по горизонталі та вертикалі. Це зазвичай кути або текстуровані області, які легко вирізнити від сусідніх фрагментів зображення. У контексті відеоаналізу транспорту, GFTT дозволяє автоматично виявити точки на транспортному засобі, що найкраще підходять для подальшого відстеження в динаміці. Наприклад, об'єктом такого відбору можуть стати краї коліс, фар, номерних знаків або інші контрастні елементи кузова.

Після виявлення таких точок, система може відстежувати їх переміщення між кадрами за допомогою алгоритмів оптичного потоку (наприклад, метода Лукаса-Канаде), що дозволяє з високою точністю оцінити зсув цих точок у просторі. Відомо, що швидкість руху транспортного засобу прямо пов'язана з темпом зміни координат цих точок у часі, тому така інформація стає основою для розрахунку швидкості. GFTT добре працює навіть у складних умовах, коли контур усього об'єкта важко виявити через часткові перекриття, розмиття або погане освітлення, але локальні контрастні ділянки залишаються придатними для аналізу.

У поєднанні з іншими методами комп'ютерного зору, GFTT може виконувати роль підготовчого етапу для побудови траєкторій або як додатковий фільтр для уточнення даних, отриманих від нейронних мереж [7]. Він забезпечує точкову, локальну оцінку руху, яку можна використати для покращення глобальних моделей трекінгу або валідації результатів, одержаних від більш комплексних систем, таких як DeepSORT. Таким чином, GFTT виступає як важливий компонент, що підвищує стійкість і надійність системи розпізнавання швидкості, особливо в умовах відеопотоку низької якості або при високій щільності трафіку.

1.3 Аналіз існуючих програмних засобів для виявлення швидкості транспортних засобів

У сфері автоматизованого визначення швидкості транспортних засобів за відео існує низка програмних рішень та наукових досліджень, які демонструють різні підходи до вирішення цього завдання. Застосовуються як класичні методи комп'ютерного зору – наприклад, оптичний потік, фіксація перетину контрольних ліній, оцінка тривалості перебування в зонах – так і сучасні підходи на основі глибокого навчання, що включають нейронні мережі для детекції та трекінгу. Цей підрозділ містить огляд таких рішень з метою виявлення їх переваг, недоліків та визначення напрямків для подальшого розвитку систем. Особлива увага приділяється відкритим дослідницьким проєктам і програмному забезпеченню з відкритим кодом, які забезпечують гнучкість і можливість адаптації до конкретних умов зйомки. На рисунку 1.5 зображено перше з таких рішень, що демонструє систему автоматизованого аналізу відеопотоку для визначення швидкості руху транспортних засобів.

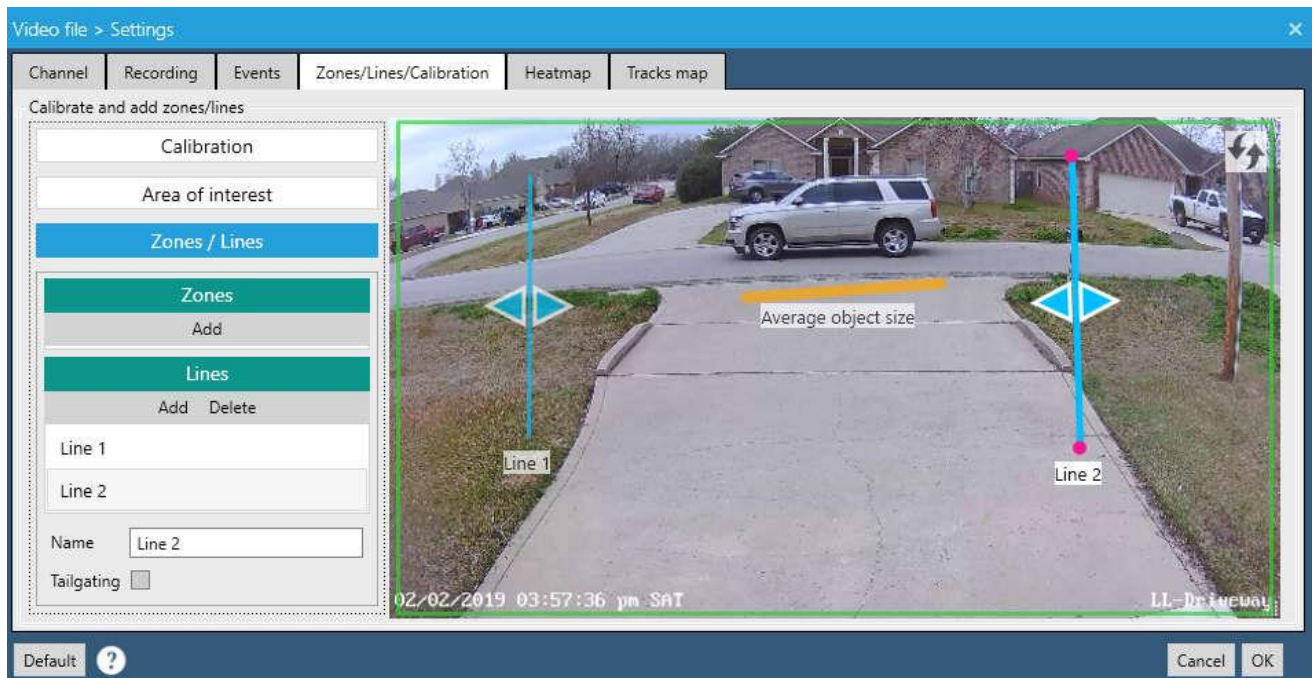


Рисунок 1.5 – Camlytics Vehicle Speed Detection [7]

Camlytics – програмне забезпечення для аналізу відео, що дозволяє визначати швидкість транспортних засобів за допомогою двох контрольних ліній

на кадрі. Система підтримує різні типи камер, включно з IP-камерами та записаними відео, має зручний інтерфейс для налаштування зон детекції та виконує обробку даних локально, що підвищує конфіденційність і не потребує підключення до мережі.

Наступним прикладом є система, описана в роботі Video-Based Vehicle Speed Estimation Using Speed Measurement Metrics. У цьому рішенні використовуються кадри відео для визначення швидкості транспортних засобів на основі аналізу відстані, пройденої між двома точками контролю. Автори роботи детально описують різні метрики вимірювання швидкості, які можуть бути застосовані залежно від умов зйомки. На рисунку 1.6 представлено користувацький інтерфейс розробленої програми, який демонструє процес детекції автомобілів, побудову траєкторії та розрахунок швидкості.

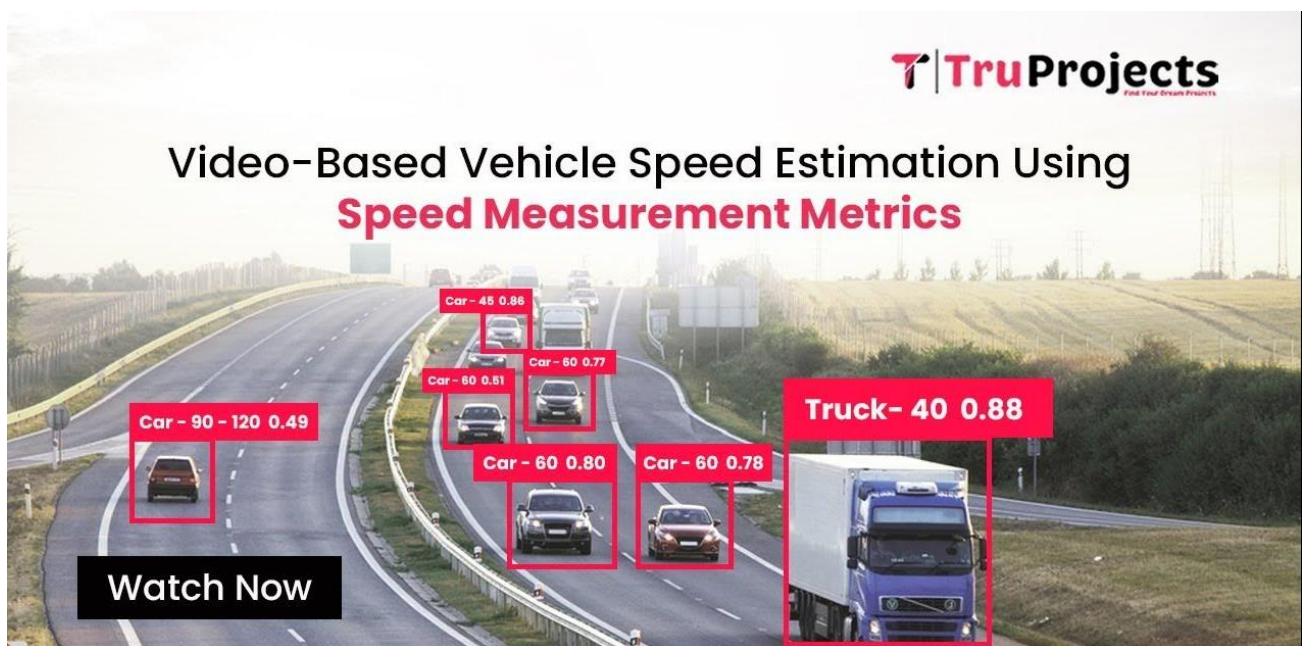


Рисунок 1.6 – Video-Based Vehicle Speed Estimation Using Speed Measurement Metrics [8]

У дослідженні Keattisak Sangsuwan та Mongkol Ekpanyarong розглядається метод визначення швидкості транспортних засобів на основі відео без використання додаткових сенсорів. Метод ґрунтується на аналізі відстані, пройденої за одиницю часу, або часу проходження відомої дистанції між контрольними лініями на зображенні. Для цього застосовано комбінацію

алгоритмів комп'ютерного зору: YOLOv3 для виявлення, DeepSORT для трекінгу, оптичний потік Лукаса-Канаде для уточнення переміщення. Система продемонструвала точність, близьку до еталонної, без потреби в апаратних засобах, що робить її перспективною для бюджетних впроваджень, хоча вона потребує ретельного калібрування та врахування зовнішніх умов.

У дослідженні, проведеному Bhushan Nikumbhe та співавторами, запропоновано систему для виявлення транспортних засобів та оцінки їх швидкості в реальному часі за допомогою обробки зображень [9]. Основним методом виявлення об'єктів є вирахування фону (background subtraction), що дозволяє ефективно виявляти рухомі об'єкти на статичному фоні. Система аналізує відеопотік, визначає транспортні засоби та розраховує їх швидкість, вимірюючи відстань, яку проходить об'єкт, за певний проміжок часу, на рисунку 1.7 зображено UI програми у процесі виконання.

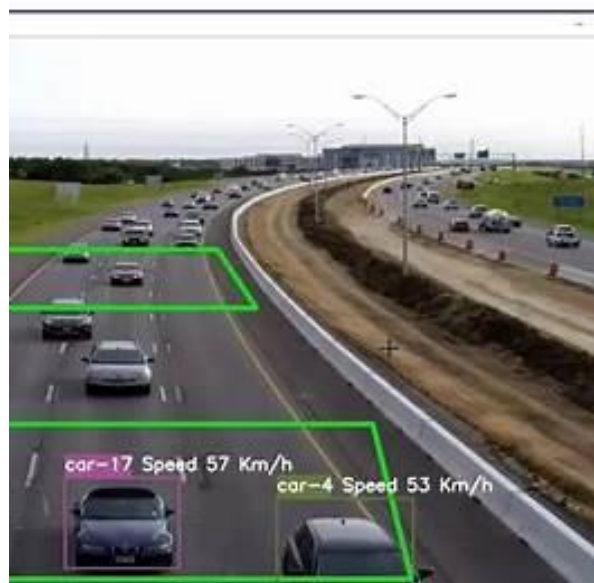


Рисунок 1.7 – Real-time Vehicle Detection and Speed Estimation System [10]

Запропонована система демонструє ефективний підхід до виявлення транспортних засобів та оцінки їх швидкості в реальному часі, використовуючи методи обробки зображень. Простота реалізації та низькі обчислювальні витрати роблять її привабливою для впровадження в існуючі системи відеоспостереження. Однак, для забезпечення високої точності необхідно враховувати вплив зовнішніх факторів та забезпечити належне калібрування системи.

Аналіз існуючих рішень показує, що хоча є ефективні методи визначення швидкості транспортних засобів за відео, багато з них мають обмеження щодо умов зйомки, необхідності точного калібрування та обмеженої адаптивності до змін у сцені. Це створює потребу в розробці більш універсальних та стійких до змін систем, які можуть працювати в різноманітних умовах без значної втрати точності.

1.4 Мета та завдання кваліфікаційної роботи

Метою кваліфікаційної роботи бакалавра є підвищення точності визначення швидкості руху автомобіля у відеопотоці засобами глибокого навчання. Запропонований підхід базується на комп'ютерному зорі та неймережевій обробці відеоданих, що дозволяє створити ефективну, доступну та безконтактну альтернативу традиційним технічним засобам контролю швидкості.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- 1) провести аналіз існуючих методів комп'ютерного зору та глибокого навчання для виявлення та трекінгу транспортних засобів у відеопотоці;
- 2) розробити метод визначення швидкості руху виявлених транспортних засобів у відеопотоці на основі координат і часових міток;
- 3) підібрати та підготувати відомий набір даних із зображень автомобілів, що включає різні умови зйомки;
- 4) створити програму реалізацію методу з підтримкою детекції, трекінгу, розрахунку швидкості та візуалізації результатів;
- 5) провести експерименти для оцінки точності запропонованого методу визначення швидкості руху автомобіля.

Успішна реалізація зазначених завдань передбачає створення повноцінної інформаційної системи, здатної автономно визначати швидкість автомобілів у відеопотоці без необхідності використання спеціалізованих сенсорів чи дорогого обладнання.

Розділ 2 Проектування методу визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами

2.1 Загальна схема методу визначення швидкості руху автомобіля у відеопотоці

У даному підрозділі представлено загальну структуру методу визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами. Метод базується на використанні технологій глибокого навчання для обробки відеоданих, що надходять із відеофайлу або з онлайн-відеопотоку. Основу процесу становить детекція транспортних засобів на послідовних кадрах відео за допомогою згорткової нейронної мережі типу YOLO (You Only Look Once), яка забезпечує високу швидкість та точність виявлення [11].

Після виявлення об'єктів система виконує їх подальше відстеження у часі за допомогою трекера DeepSORT, що дозволяє встановити ідентичність об'єкта між кадрами та побудувати його повну траєкторію. Це дає змогу визначити зміщення об'єкта між кадрами, а знаючи частоту відео (FPS) та параметри сцени (наприклад, розміри зони спостереження у метрах), можна розрахувати швидкість об'єкта у фізичних одиницях – метрах за секунду або кілометрах на годину [12].

Метод передбачає поетапну обробку вхідного відео: починаючи з конфігурації параметрів камери, масштабування сцени та побудови зон перспективи, і закінчуючи фільтрацією об'єктів за класом, визначенням моменту перетину контрольної лінії та обчисленням швидкості з урахуванням просторових зон та поправок. Такий підхід дозволяє не лише забезпечити високу точність у прогнозах, але й адаптувати систему до різних дорожніх умов та конфігурацій камери.

Схематичне зображення логіки функціонування методу подано на рисунку 2.1.

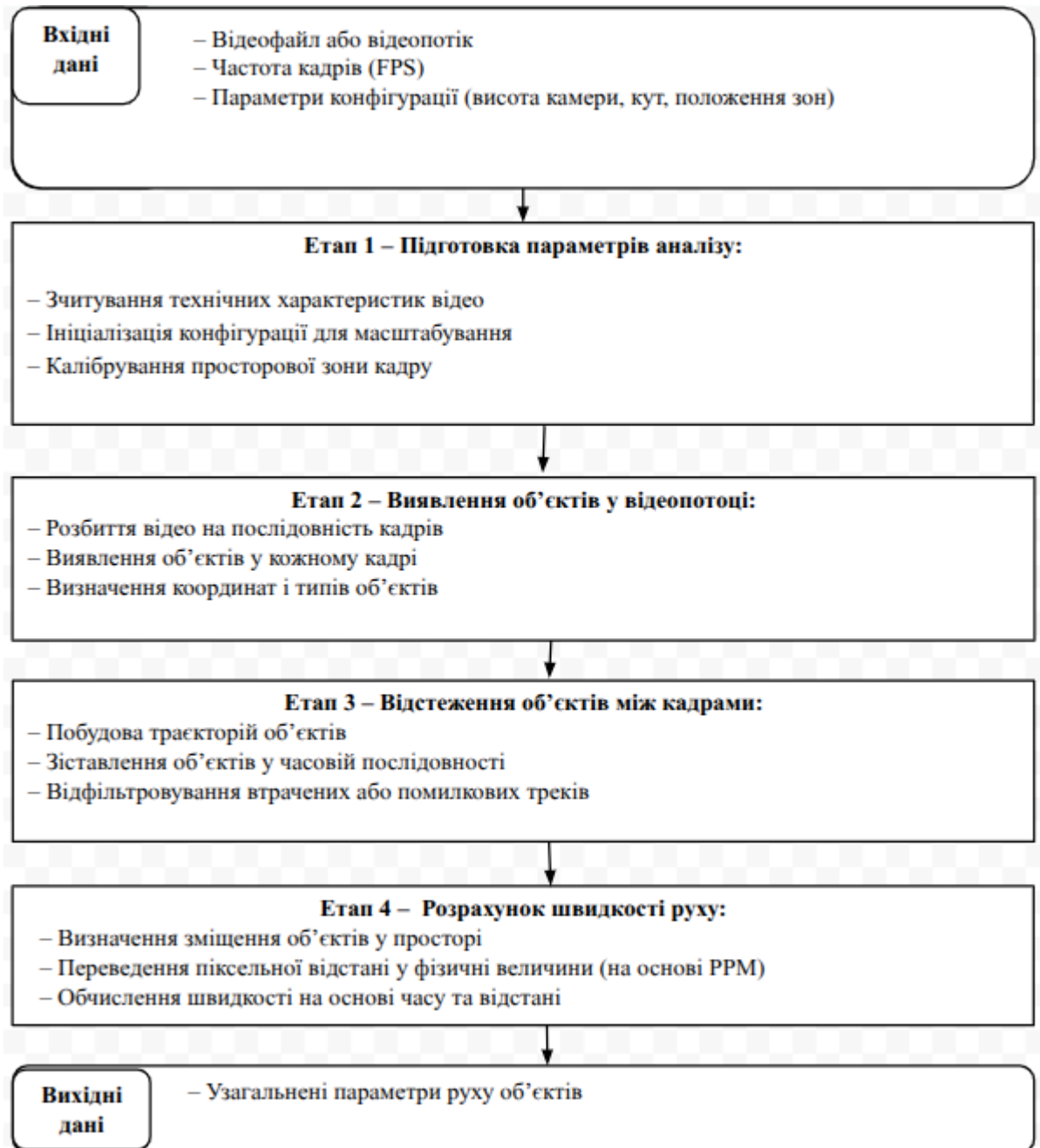


Рисунок 2.1 – Загальна схема методу визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами

Розробка методу визначення швидкості транспортних засобів у відеопотоці базується на поетапному аналізі відеоданих, який охоплює чотири логічно послідовних стадій. Вхідними даними для системи є відеозапис, параметри конфігурації камери (висота встановлення, кут нахилу, положення зон

спостереження), а також частота кадрів (FPS), що дозволяє забезпечити коректну часову прив'язку об'єктів у просторі.

На першому етапі здійснюється попереднє налаштування параметрів аналізу. Зокрема, зчитуються технічні характеристики відеофайлу: роздільна здатність, частота кадрів (FPS), тривалість тощо. Один із ключових параметрів – коефіцієнт перетворення пікселів у метри (PPM), який розраховується на основі відомої довжини об'єкта чи відстані у кадрі. Цей коефіцієнт дозволяє здійснювати перехід від піксельної координатної системи до метричної, що необхідно для точного обчислення швидкості.

Другий етап присвячено виявленню транспортних засобів у кожному кадрі відеопотоку. Відео розбивається на окремі кадри з відповідною часовою розміткою. Для кожного кадру виконується інференція за допомогою попередньо навченого детектора об'єктів YOLOv10n. Результатом цього етапу є набір детекцій із вказаними координатами прямокутних рамок (bounding boxes), класами об'єктів (зокрема – «car») і рівнями впевненості. Детекції, що не проходять заданий поріг довіри, відсіюються. Таким чином формується структурований набір об'єктів, що потенційно підлягають подальшому аналізу.

На третьому етапі відбувається відстеження об'єктів у часовому просторі – трекінг. Для цього використовується алгоритм DeepSORT, який дозволяє зіставити детекції між сусідніми кадрами, спираючись на візуальні ознаки (реідентифікаційні вектори) та просторову близькість об'єктів. Результатом є побудовані унікальні траєкторії руху кожного транспортного засобу, які зберігаються як послідовності координат із часовими мітками. Під час цього етапу відбувається також фільтрація нестійких або фрагментованих треків, що можуть бути викликані частковими перекриттями чи помилками в детекції.

Четвертий етап полягає у визначенні швидкості руху для кожного транспортного засобу. Зміщення об'єкта між кадрами розраховується як евклідова відстань між координатами центру об'єкта у двох різних моментах часу. Це зміщення, виражене в пікселях, ділиться на часовий інтервал між відповідними кадрами (визначається через FPS) і множиться на коефіцієнт PPM, що дозволяє

отримати швидкість у метрах за секунду. Для уникнення різких стрибків у значеннях швидкості внаслідок шумів чи нестабільної детекції, результати згладжуються за допомогою фільтра Савицького-Голея. У результаті формується достовірна крива швидкості об'єкта у часі.

2.2 Архітектура нейромережевої моделі для обробки відеопотоку

YOLOv10 – це сучасна архітектура нейронної мережі, призначена для задач виявлення об'єктів у реальному часі. Вона поєднує ефективність обчислень із високою точністю, що робить її придатною для застосування в мобільних та вбудованих системах. Архітектура підтримує багаторівневе навчання, що дозволяє ефективно виявляти об'єкти різного масштабу. Нижче представлена схема моделі YOLOv10:

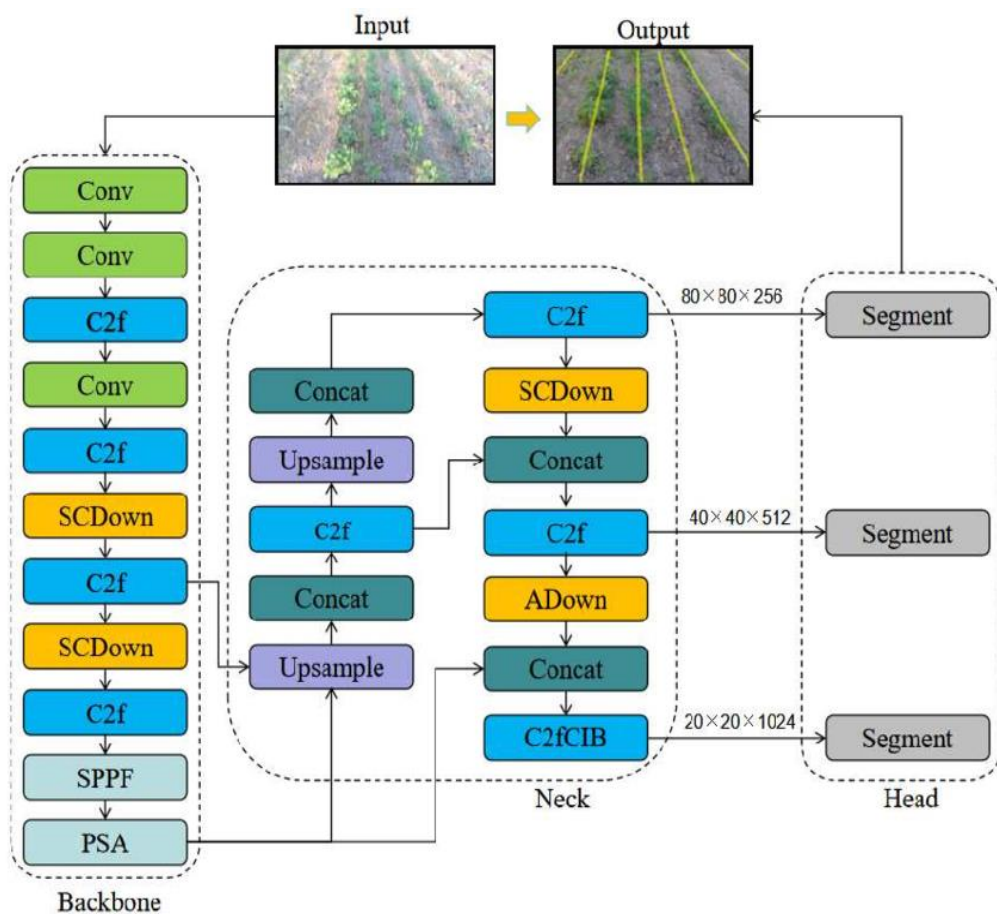


Рисунок 2.2 – Архітектура моделі YOLOv10 [13]

Модель YOLOv10 складається з трьох основних компонентів: Backbone, Neck і Head, кожен із яких виконує окрему функцію в процесі обробки зображення для виявлення об'єктів.

Backbone є ключовою частиною архітектури YOLOv10, відповідальною за витягнення ознак із вхідного зображення. Його основна мета – побудувати багаторівневе представлення об'єкта, що дозволяє моделі ефективно розпізнавати об'єкти різних розмірів і форм. Структура backbone починається зі згорткових шарів (Conv), які виконують базове оброблення зображення, витягуючи низькорівневі ознаки та поступово зменшуючи просторову роздільну здатність [14]. Далі використовуються блоки C2f (Cross-Stage Partial Fusion), які забезпечують ефективну передачу інформації між шарами з мінімальними втратами, комбінуючи глибокі й поверхневі ознаки. Для кращого збереження просторової інформації та підвищення глибини мережі застосовуються SCDown (Spatial-Channel Downsampling) – модулі, що об'єднують як просторове зменшення, так і каналову агрегацію, покращуючи узагальнюючу здатність моделі. У нижній частині backbone розміщено SPPF (Spatial Pyramid Pooling Fast) – компонент, який дозволяє агрегувати ознаки з різних масштабів, що особливо важливо для виявлення об'єктів з різною геометрією. Нарешті, PSA (Parameter-free Self-Attention) додає шар самоуваги без додаткових параметрів, дозволяючи моделі акцентувати увагу на ключових зонах зображення без ускладнення структури. У результаті, backbone формує глибоке багаторівневе представлення вхідного зображення, що охоплює як дрібні деталі, так і глобальні контекстуальні зв'язки, необхідні для подальшого виявлення об'єктів.

Neck у структурі YOLOv10 виконує функцію проміжної обробки ознак, поєднуючи інформацію з різних рівнів, що була отримана на етапі backbone. Його головне завдання – збагачення просторово-контекстуального представлення об'єктів перед передачею їх у вихідну частину (head) для остаточного розпізнавання [15]. Першим важливим компонентом neck є Upsample – операція, яка дозволяє збільшити роздільну здатність ознак, отриманих із глибших шарів, і поєднати їх із ознаками вищої роздільності з попередніх рівнів. Завдяки цьому

модель зберігає як глобальну, так і локальну інформацію. Після цього застосовується операція Concat, яка об'єднує ознаки з різних шляхів, забезпечуючи комплексне контекстуальне представлення. Об'єднані ознаки надходять у блок C2f, який дозволяє моделі краще узагальнювати отримані характеристики та ефективно їх обробляти. Додатково використовується ADown – модуль адаптивного зменшення розмірності, що дозволяє вибірково зберігати найбільш релевантні ознаки з обмеженою втратою інформації. На завершення, застосовується блок C2fCIB – вдосконалений варіант C2f, який містить механізми збереження ключової інформації з вищих рівнів мережі та покращує розподіл ваг між каналами. У результаті, neck формує три набори ознак з роздільністю $80 \times 80 \times 256$, $40 \times 40 \times 512$ та $20 \times 20 \times 1024$, які передаються в голову моделі для остаточного розпізнавання об'єктів на різних масштабах. Такий підхід дозволяє досягти високої точності виявлення об'єктів незалежно від їх розміру.

Head є фінальним етапом обробки в YOLOv10 і відповідає за генерацію остаточних передбачень на основі ознак, сформованих у neck. Для кожного з трьох масштабів ознак ($80 \times 80 \times 256$, $40 \times 40 \times 512$, $20 \times 20 \times 1024$) окремо виконується детекція або сегментація об'єктів. В залежності від задачі, відповідний блок може реалізовувати функціонал Segment Head (для сегментації) або Detect Head (для виявлення об'єктів). Саме тут розраховуються координати меж об'єкта (bounding box), клас об'єкта та рівень довіри моделі щодо правильності передбачення [16]. Таким чином, head трансформує багатовимірні ознаки в чіткі вихідні результати – набір виявлених об'єктів із визначеними координатами та класовою належністю, які можна використовувати в прикладних завданнях, як-от моніторинг трафіку або підрахунок об'єктів.

YOLOv10 має низку переваг, які роблять його ефективним вибором для задач комп'ютерного зору. Завдяки глибокій ф'юзії ознак на всіх рівнях модель забезпечує високу точність виявлення як великих, так і дрібних об'єктів. Застосування оптимізованих і легких блоків, зокрема PSA (Parameter-free Self-Attention), сприяє зменшенню обчислювальних витрат і підвищенню швидкості. Архітектура YOLOv10 є масштабованою та придатною до розгортання на

пристроях із обмеженими ресурсами, включно з мобільними платформами. Крім того, підтримка мультизадачності – наприклад, одночасної детекції та сегментації – реалізована завдяки універсальній будові head, що дає змогу адаптувати модель до різних типів комп’ютерного зору.

2.3 Підготовка даних для донавчання моделі

Для виявлення транспортних засобів у відеопотоці застосовано спеціалізований датасет DAWN із високою точністю анотацій. Відеозаписи дорожнього руху відібрано з відкритих джерел і камер спостереження, охоплюючи широкий спектр умов навколишнього середовища, зокрема різноманітні погодні умови (туман, дощ, сонячна погода), зміни освітлення (денне та нічне відео, тіні) та різні ракурси зйомки (фронтальний, діагональний, з висоти) [17].

На рисунку 2.3 зображено розподіл зображень у датасеті.

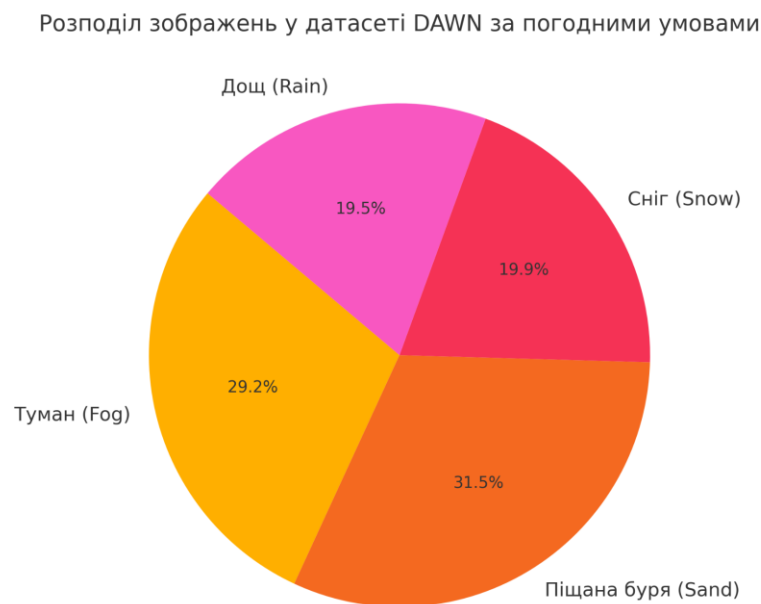


Рисунок 2.3 – Розподіл зображень у датасеті DAWN за погодними умовами [17]

Кожне зображення містить один або кілька анотованих автомобілів, що забезпечує підготовку моделі до виявлення транспортних засобів у складних погодних умовах.

На рисунку 2.4 представлено приклад зображення.



Рисунок 2.4 – Приклад зображення для датасету

Приклад кадру з датасету зафіксовано в умовах густого туману. Незважаючи на знижений рівень видимості, на зображенні чітко видно фари автомобілів, що є важливим для тестування моделі в складних умовах навколишнього середовища.

Відеозаписи розбито на окремі зображення. Усі зображення проанотовано вручну у середовищі Roboflow – спеціалізованій платформі для комп'ютерного зору, яка дозволяє створювати та експортувати датасети у форматі YOLO [18]. На рисунку 2.5 зображено приклад анотації.

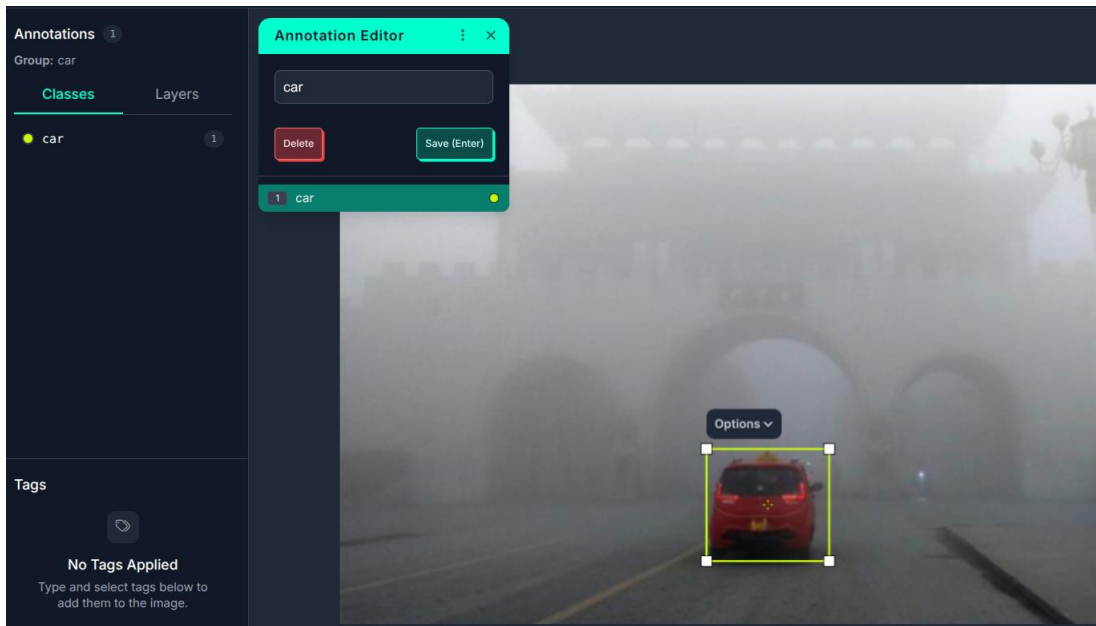


Рисунок 2.5 – Приклад анотування зображення для датасету в Roboflow

Структура датасету приведено до формату, сумісного з бібліотекою Ultralytics: папки `images/train` і `images/val` для зображень, `labels/train` і `labels/val` для анотацій у форматі YOLO, та файл `data.yaml` для конфігурації. Датасет поділено на навчальну (80 %) та валідаційну (20 %) вибірки. Така організація забезпечує коректне навчання моделі, а також її перевірку під час валідації та тестування [19]. Навчальна вибірка застосована для тренування моделі, валідаційна – для перевірки узагальнення та підбору параметрів.

Для донавчання моделі YOLOv10n, попередньо натренованої на датасеті COCO, застосовано підхід *transfer learning*, орієнтований на адаптацію детектора до цільової категорії «автомобіль». Навчання проводиться протягом 50 епох на зображеннях розміром 640×640 пікселів із використанням оптимізатора AdamW та розміром пакета (*batch size*) 16 [20]. З метою стабілізації навчання на початкових етапах реалізовано режим розігріву (*warmup*) протягом кількох перших епох.

Для покращення здатності моделі до узагальнення застосовуються базові методи аугментації, зокрема випадкове горизонтальне віддзеркалення, масштабування та варіації яскравості [21]. Навчання виконується з використанням

сучасного графічного процесора, наприклад NVIDIA RTX 4060 або іншого з аналогічною обчислювальною потужністю [22].

Запропоновано використання наперед навченої моделі YOLO, яка має базову здатність до виявлення загальних об'єктів. У рамках цього підходу пропонується проектування процесу донавчання, який адаптує модель до спеціалізованого датасету, що містить лише транспортні засоби [23]. Такий підхід дозволяє зосередити обчислювальні ресурси на цільовому класі «автомобіль» і забезпечити підвищену точність детекції в конкретних умовах.

Проектована процедура передбачає оптимізацію ваг моделі за допомогою стандартних функцій втрат. Зокрема, функція IoU loss (Intersection over Union) використовується для підвищення точності просторового окреслення об'єктів (bounding boxes). Функція classification loss, наприклад крос-ентропія, забезпечує точнішу класифікацію об'єктів, тоді як objectness loss відповідає за визначення ймовірності наявності об'єкта в кадрі [24].

Кількість епох встановлюється в межах 10–15 для забезпечення балансу між ефективністю навчання та уникненням перенавчання. У разі недостатньої якості розпізнавання передбачено можливість зміни кількості епох. Для підвищення узагальнюючої здатності моделі застосовуються базові методи аугментації даних, зокрема горизонтальне віддзеркалення, масштабування та зміна яскравості.

Результатом процесу донавчання є адаптована модель YOLO, спроектована для виявлення автомобілів на спеціалізованому наборі зображень з різними умовами освітлення, погодними факторами та ракурсами зйомки.

2.4 Виявлення об'єктів у відеопотоці

Детекція автомобілів у межах проектування методу визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами є вихідною точкою для подальших обчислень.

На рисунку 2.6 представлено спроектовану схему виявлення автомобілів у відеопотоці за допомогою нейромережі YOLO.

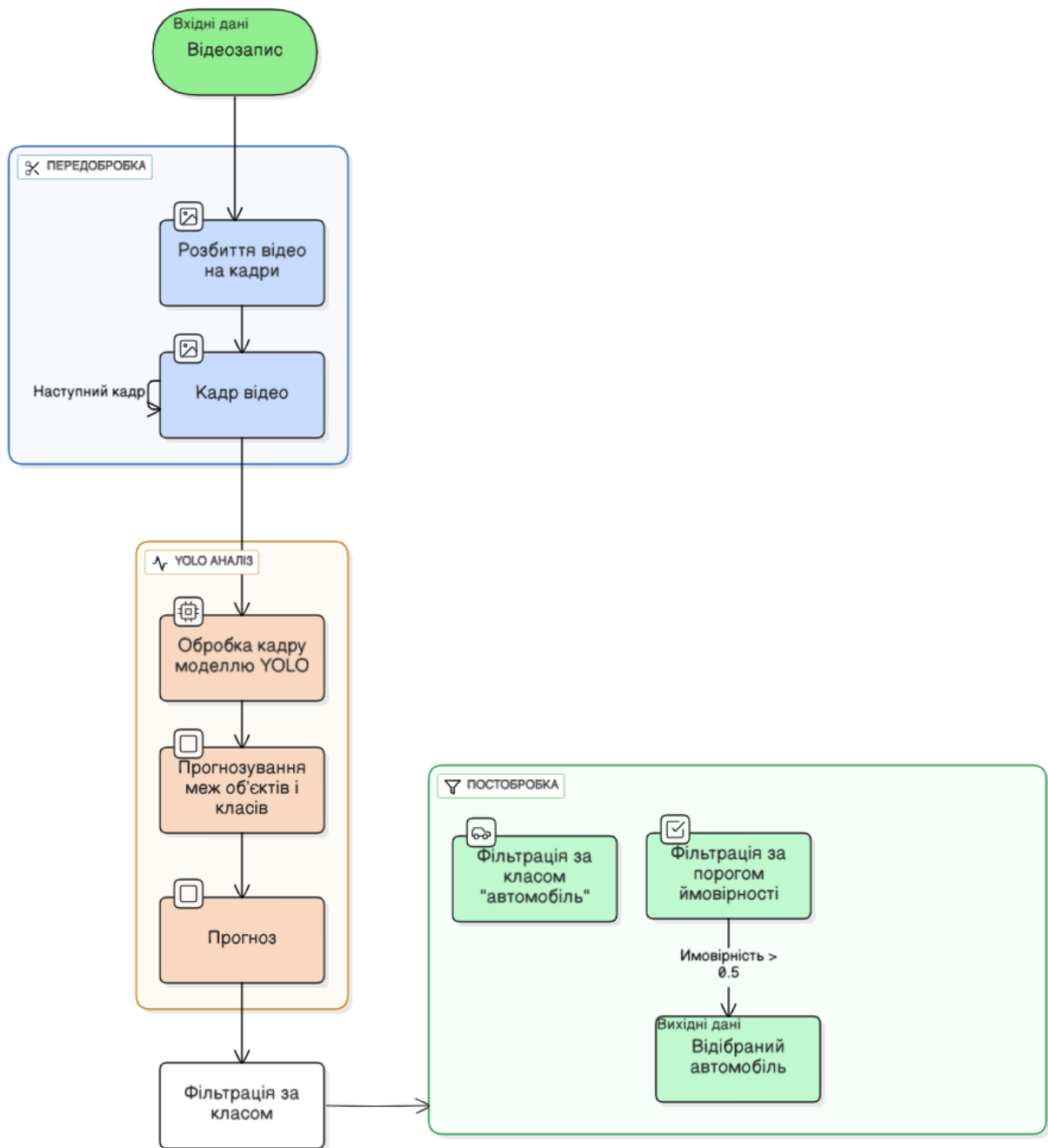


Рисунок 2.6 – Схема виявлення автомобілів у відеопотоці за допомогою нейромережі YOLO

Схема на рисунку ілюструє етап виявлення автомобілів у відеопотоці з використанням нейромережі YOLO. Вона поділена на три логічні частини: передобробка, аналіз YOLO та постобробка.

На вході подається відео, яке спочатку розбивається на окремі кадри (передобробка). Кожен кадр передається на обробку нейронною мережею YOLO, яка виконує прогнозування: визначає межі об'єктів (bounding boxes) і класи для кожного виявленого об'єкта. Цей процес представлено у блоці «YOLO аналіз».

Після отримання прогнозів виконується постобробка. Результати фільтруються: спершу залишаються лише об'єкти, що класифіковані як «автомобіль», а потім – ті, ймовірність виявлення яких перевищує заданий поріг (наприклад, 0.5). Відібрані автомобілі передаються на подальші етапи аналізу або виводу. Така схема забезпечує виявлення тільки релевантних об'єктів у відеопотоці.

2.5 Визначення швидкості руху автомобілів

Визначення швидкості руху транспортних засобів у даному методі базується на аналізі результатів детекції, отриманих за допомогою нейромережевої моделі YOLO (You Only Look Once). Основними вихідними даними є координати центрів обмежувальних прямокутників (bounding boxes), їхні розміри та класи об'єктів. Ці координати дозволяють визначити положення кожного транспортного засобу на кожному кадрі відео. У подальшому, шляхом відстеження зміщення центру об'єкта між послідовними кадрами, розраховується пройдена відстань у пікселях. Після перетворення цієї відстані у фізичні одиниці за допомогою коефіцієнта масштабу (PPM – pixels per meter) та врахування часового інтервалу між кадрами, система обчислює швидкість об'єкта [25]. На рисунку 2.7 представлено схематичну реалізацію цього підходу на основі даних детекції.

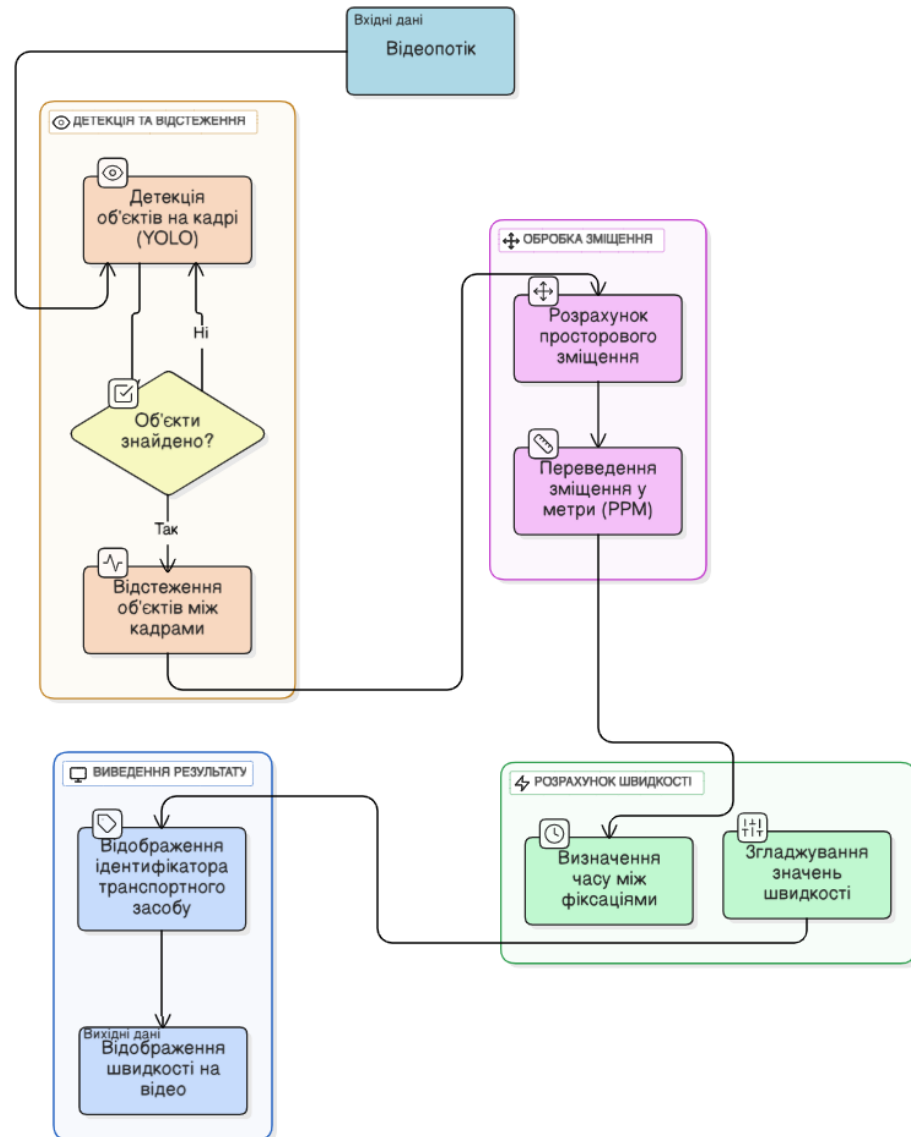


Рисунок 2.7 – Схема обчислення швидкості автомобілів у відеопотоці за допомогою неймережі YOLO

На кожному кадрі відеопотоку виконується детекція транспортних засобів. Для кожного об'єкта модель формує прямокутник із координатами (x_1, y_1, x_2, y_2) . Центр об'єкта визначається за координатами прямокутника:

$$x_c = \frac{x_1 + x_2}{2}, y_c = \frac{y_1 + y_2}{2} \quad (2.1)$$

де x_1, y_1 – координати лівого верхнього кута, а x_2, y_2 – координати правого нижнього кута bounding box [26].

Після детекції транспортних засобів на кожному кадрі відеопотоку необхідно забезпечити відстеження їхнього переміщення між послідовними

кадрами. Це дає змогу зіставляти положення одного й того самого об'єкта в часі та розраховувати його зміщення, що є основою для визначення швидкості руху.

Для цього кожному об'єкту присвоюється унікальний ідентифікатор, який зберігається протягом його присутності в кадрі. Таким чином, система може «розпізнати» автомобіль на наступному кадрі як той самий, що був виявлений раніше, навіть якщо його координати змінилися. У проєкті для цього використовується вбудований трекер DeepSORT, інтегрований у модель YOLOv10 через метод `.track()` з опцією `persist` [27]. Це дає змогу автоматично зберігати ідентифікатори об'єктів між кадрами, що необхідно для оцінки переміщення і розрахунку швидкості.»

Для кожного об'єкта зберігаються поточні координати центру, час останньої фіксації, історія обчислених швидкостей (для подальшого згладжування), а також ознака проходження контрольної лінії. Коли об'єкт перетинає задану лінію, система виконує обчислення його швидкості на основі пройденої відстані та часу, що минув з попереднього положення.

Цей підхід дозволяє точно визначати швидкість руху окремих автомобілів у реальному часі, незалежно від кількості транспортних засобів у кадрі.

Для кожного об'єкта фіксується його положення на двох послідовних кадрах відео: на попередньому кадрі $P_{t-1} = (x_{t-1}, y_{t-1})$ на поточному кадрі – $P_t = (x_t, y_t)$. Просторове зміщення між цими двома положеннями визначається як евклідова відстань між точками у площині зображення:

$$\Delta S_{px} = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (2.2)$$

де (x_t, y_t) – координати об'єкта на поточному кадрі, (x_{t-1}, y_{t-1}) – координати на попередньому кадрі [28].

Після того як просторове зміщення об'єкта було визначено у пікселях, виникає потреба перетворити його у метри, щоб надалі можна було коректно розраховувати фізичну швидкість. Це перетворення здійснюється за допомогою коефіцієнта масштабу, який позначається як PPM – кількість пікселів на один метр. Застосування цього коефіцієнта дозволяє встановити відповідність між

пiксельною вiдстанню на зображеннi та реальним лiнiйним змiщенням у фiзичному просторi.

Однак значення PPM не є постійним по всьому кадру. Воно залежить вiд ряду чинникiв, серед яких найважливишими є координати об'єкта на зображеннi, тобто його вертикальне та горизонтальне положення. Через ефект перспективи об'єкти, що розташованi ближче до верхньої частини зображення, здаються меншими, а вiдповiдно й один метр у реальностi займає менше пiкселiв. Горизонтальне положення також впливає, особливо у випадках, коли камера спрямована пiд кутом або покриває кiлька смуг руху, що мають рiзну глибину огляду [29].

Додатково, на значення PPM впливають параметри самої камери, зокрема її висота над землею, кут нахилу, фокусна вiдстань та розмiр матрицi. Усi ці фактори враховуються при побудовi карти перспективних зон, яка створюється пiд час попередньої калiбровки сцени [30]. Саме з цієї карти система зчитує актуальне значення коефiцiєнта PPM для конкретної координати об'єкта. Завдяки цьому можна точно визначити, яку фiзичну вiдстань подолав транспортний засiб, гiрнтуючись на його змiщеннi в пiксельному просторi. Це, у свою чергу, дозволяє виконати подальший етап – обчислення швидкостi.

Для переведення пiксельної вiдстанi в метри:

$$\Delta S_m = \frac{\Delta S_{px}}{PPM(x,y)} \quad (2.3)$$

де ΔS_{px} – пiксельне змiщення, $PPM(x, y)$ – коефiцiєнт пiкселiв на метр у точцi (x, y) .

Пiсля того як реальна вiдстань, яку подолав транспортний засiб мiж двома кадрами, була обчислена в метрах, стає можливим розрахунок його швидкостi. Для цього необхідно знати також часовий iнтервал, що пройшов мiж фiксацiєю об'єкта на попередньому та поточному кадрах. Зазвичай цей iнтервал вiдповiдає частотi кадрiв вiдео, тобто становить сталу величину, яка визначається характеристиками вiдеофайлу.

Швидкiсть визначається як вiдношення пройденої вiдстанi до часу. Оскiльки об'єкти перемiщуються по зображенню у метрах, а результат необхідно

подати у кілометрах на годину, додатково застосовується коефіцієнт перерахунку одиниць. Формула виглядає наступним чином:

$$v = \frac{\Delta s_m}{\Delta t} \cdot 3.6 [\text{км/год}], \quad (2.4)$$

де:

Δs_m – пройдена відстань об'єкта в метрах;

Δt – часовий інтервал між кадрами, що визначається частотою відео (наприклад, при 30 FPS: $\Delta t = \frac{1}{30}$);

3.6 – коефіцієнт перерахунку з м/с у км/год.

Під час обробки відеопотоку координати виявлених об'єктів можуть незначно змінюватися навіть у випадку їх рівномірного руху. Такі флуктуації можуть бути спричинені дрібними похибками детекції, шумом зображення або нестабільністю трекара, що призводить до стрибків у розрахованих значеннях швидкості. Щоб уникнути цих коливань і отримати стабільніші результати, застосовується згладжування отриманих значень швидкості.

У даній системі використовується метод згладжування Савицького-Голея (Savgol Filter), який дозволяє зберігати загальний тренд зміни швидкості, одночасно зменшуючи вплив випадкових відхилень [31]. Замість того, щоб використовувати миттєве значення швидкості, система розраховує згладжене значення на основі декількох попередніх вимірювань. Формально це можна подати у вигляді:

$$v_{\text{згладжене}} = SG(v_1, v_2, \dots, v_N) \quad (2.5)$$

де v_1, v_2, \dots, v_N – послідовність миттєвих швидкостей.

Розрахована швидкість накладається безпосередньо на відео над відповідним транспортним засобом. Формат виводу інформації є простим і зрозумілим: поряд із кожним об'єктом відображається його унікальний ідентифікатор та обчислене значення швидкості у кілометрах на годину. Наприклад [17] – [45 км/год]

Цей підпис дозволяє швидко оцінити швидкість руху кожного автомобіля у кадрі під час перегляду відео в реальному часі або в записі. Завдяки інтеграції

обробки візуальної інформації з аналітичними обчисленнями, система забезпечує зручний та інформативний спосіб представлення результатів.

2.6 Метрики оцінювання якості моделі

У процесі оцінювання якості моделі детекції автомобілів у відеопотоці використовуються низка класичних метрик, що дозволяють кількісно характеризувати якість роботи нейромережевого детектора. Ассурасу визначає загальну частку об'єктів, які були правильно класифіковані як автомобілі серед усіх об'єктів, виявлених у відео. Проте ця метрика є чутливою до дисбалансу класів – зокрема, у випадках, коли у вибірці переважає фон, вона може бути надмірно оптимістичною. Precision, своєю чергою, показує, яка частка виявлених як автомобілі об'єктів насправді є автомобілями, тобто характеризує точність виявлення та дозволяє оцінити рівень помилкових спрацьовувань. Recall відображає здатність моделі знаходити всі об'єкти класу "автомобіль", що реально присутні в кадрах, і чим вищий цей показник, тим менше об'єктів залишилося непоміченими. Щоб збалансувати обидва підходи, застосовується F1-score – гармонійне середнє між precision та recall, яке дозволяє отримати інтегральну оцінку загальної якості моделі, особливо в умовах, коли обидві характеристики однаково важливі.

F1-score обчислюється за формулою:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.6)$$

Сукупне використання цих метрик забезпечує повну та об'єктивну картину якості системи виявлення транспортних засобів, для візуального аналізу моделі може застосовуватися ROC-крива, приклад на рисунку 2.8.

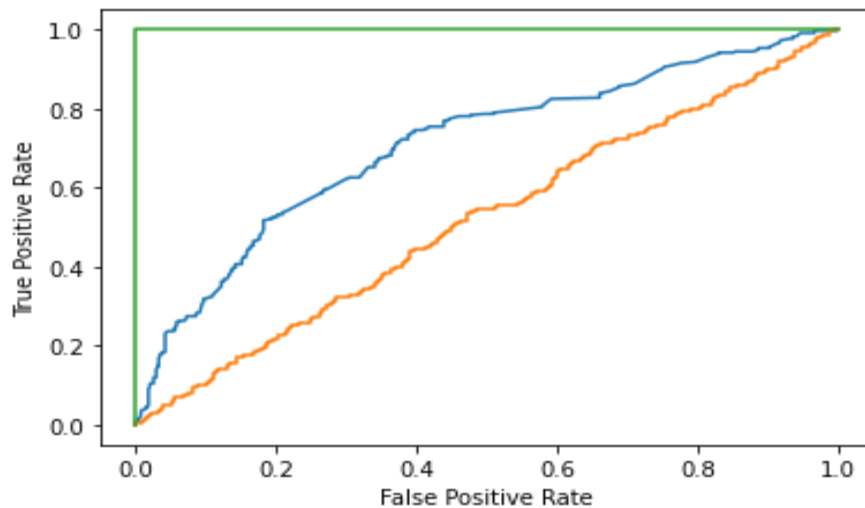


Рисунок 2.8 – Приклад графіку ROC-крива [32]

ROC-крива (Receiver Operating Characteristic) є важливим інструментом для оцінки продуктивності класифікатора. Вона відображає залежність між часткою правильно виявлених позитивних об'єктів (True Positive Rate) і часткою хибнопозитивних спрацьовувань (False Positive Rate) при варіації порогу впевненості класифікації. Крива ілюструє, як змінюється здатність моделі виявляти цільовий клас (у даному випадку – автомобілі) залежно від того, наскільки «вимогливим» є поріг. Ідеальний класифікатор матиме ROC-криву, яка проходить через верхній лівий кут графіка, що відповідає максимальному True Positive Rate за нульового False Positive Rate. Площа під ROC-кривою (AUC – Area Under Curve) використовується як числова метрика: чим вона ближча до 1, тим краща якість моделі, наступний приклад візуального аналізу моделі зображено на рисунку 2.9.

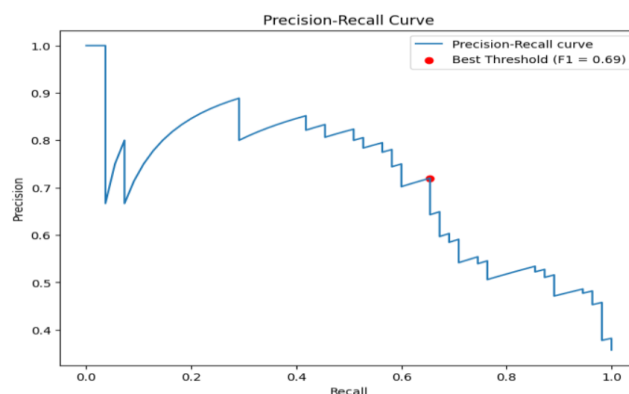


Рисунок 2.9 – Приклад графіку Precision-Recall крива [33]

Precision-Recall крива – це графічне представлення співвідношення між точністю (precision) та повнотою (recall) моделі при різних порогах впевненості. На відміну від ROC-кривої, вона є особливо інформативною в умовах значного дисбалансу класів, коли цільовий об’єкт (наприклад, автомобіль) зустрічається набагато рідше, ніж фонові елементи. Precision-Recall крива дозволяє оцінити, як добре модель справляється з виявленням рідкісних об’єктів, показуючи, наскільки велика частка знайдених об’єктів справді належить до цільового класу. Чим більша площа під кривою (AP – average precision), тим вища загальна якість детекції, наступний приклад візуального аналізу моделі зображено на рисунку 2.10.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Рисунок 2.10 – Приклад графіку Матриця помилок [33]

Confusion Matrix – це таблиця, яка наочно відображає кількість правильних і помилкових класифікацій, здійснених моделлю. Вона містить чотири основні категорії: true positives (TP) – об’єкти, правильно класифіковані як автомобілі; false positives (FP) – об’єкти, помилково визначені як автомобілі; true negatives (TN) – об’єкти, правильно розпізнані як неавтомобілі; false negatives (FN) – автомобілі, які модель не змогла виявити. Така матриця дозволяє швидко і наочно проаналізувати помилки моделі та зрозуміти, чи модель більше схильна до

пропусків (FN) чи до хибних спрацьовувань (FP), що особливо важливо при налаштуванні порогів детекції або підборі додаткових критеріїв фільтрації.

Усі ці метрики й графіки дозволяють не лише кількісно, а й візуально оцінити продуктивність моделі після її донавчання та адаптації до конкретного датасету автомобілів.

Висновки до розділу 2

Запропонований метод є багатокomпонентним і охоплює кілька ключових етапів обробки відеоданих, які включають попередню конфігурацію параметрів сцени, детекцію об'єктів за допомогою нейронної мережі YOLOv10, трекінг за алгоритмом DeepSORT, обчислення піксельного зміщення та його перетворення у швидкість руху на основі метричних коефіцієнтів.

Особливу увагу під час проєктування приділено модулю масштабування, що забезпечує перетворення координат з пікселів у метри з урахуванням перспективної геометрії сцени, положення об'єкта в кадрі, висоти та кута встановлення камери. Розрахунок коефіцієнта PPM (pixels per meter) здійснюється з врахуванням локалізації об'єкта, що дозволяє зменшити похибку при визначенні фактичної відстані, пройденої автомобілем між кадрами. Також був передбачений механізм згладжування отриманих значень швидкості методом Савицького-Голея, що дозволяє знизити вплив флуктуацій, викликаних шумом в детекції або трекінгу.

В межах розділу детально розглянуто архітектуру моделі YOLOv10. Описано процес донавчання моделі на відомому датасеті з урахуванням особливостей дорожнього середовища, зокрема погодних умов, різних ракурсів камер та типів транспортних засобів. Пояснено, як здійснюється підготовка даних, а також як організовано розподіл на навчальну та валідаційну вибірки.

Наприкінці розділу представлено набір основних метрик оцінки якості методу, таких як Precision, Recall, F1-score, MAE, RMSE, а також способи їхньої візуалізації через PR-криві, ROC-криві та матриці помилок.

Розділ 3 Програмна реалізація методу визначення швидкості руху автомобіля у відеопотоці нейромеревевими засобами

3.1 Засоби розробки програмного застосунку

Для розробки програмного застосунку, призначеного для визначення швидкості руху транспортних засобів у відеопотоці, як основну мову програмування обрано Python. Вона є високорівневою та широко застосовується у сфері комп'ютерного зору й глибокого навчання завдяки своїй простоті використання та потужній екосистемі бібліотек [34].

Передбачено використання бібліотеки OpenCV для обробки відео та зображень, зокрема доступу до окремих кадрів, нанесення графічних елементів (bounding boxes, текст) і виконання базових операцій попередньої обробки. Інференція нейромережевої моделі YOLOv10n здійснюється за допомогою бібліотеки Ultralytics YOLO, яка також забезпечує базовий трекінг об'єктів за допомогою функції track().

Для реалізації чисельних розрахунків, зокрема визначення евклідової відстані між положеннями транспортного засобу, а також згладжування швидкості, передбачено використання бібліотек NumPy і SciPy [35].

Графічний інтерфейс користувача створюється з використанням бібліотеки Tkinter, що забезпечує можливість взаємодії з усіма параметрами аналізу відео без необхідності редагування коду [36].

Модель YOLOv10n передбачено донавчити на спеціалізованому датасеті із фокусом на клас «автомобіль», що дозволяє забезпечити точну детекцію в умовах реального середовища, включаючи туман, дощ і складне освітлення. Інтеграція зазначених інструментів дозволяє сформувати технічну основу для побудови повнофункціональної системи аналізу відеопотоку з метою визначення швидкості транспортних засобів.

3.2 Структура та особливості реалізації програмного застосунку

Програмний застосунок побудовано за модульною структурою, що забезпечує чітке розділення відповідальностей між функціональними компонентами системи. Основу реалізації становить головне вікно конфігуратора, розроблене на базі бібліотеки Tkinter, яке дозволяє користувачу взаємодіяти з усіма параметрами аналізу відео через зручний графічний інтерфейс. Компонування реалізовано у вигляді окремих блоків: вибір відеофайлу, конфігурація параметрів камери, зон та смуг, корекція спотворень, параметри візуалізації, а також кнопки керування (збереження/завантаження конфігурації та запуск аналізу).

Вся логіка обробки відео винесена в окрему функцію `process_video_with_config`, яка отримує конфігурацію з інтерфейсу та виконує повний цикл: завантаження відео, детекція об'єктів за допомогою YOLOv10n, трекінг за допомогою вбудованого механізму `YOLO.track()`, розрахунок швидкості з урахуванням зони та смуги, згладжування швидкості і накладання результатів на відеокадри в реальному часі. Для гнучкості система підтримує перспективну прив'язку коефіцієнта масштабу PPM залежно від координат об'єкта та визначеної смуги.

Особливістю реалізації є підтримка відображення зон перспективи, конфігурування меж смуг руху, компенсація спотворень лінзи камери та можливість зберігати параметри в JSON-конфігурації для подальшого повторного використання [37]. Всі параметри – включно з висотою камери, кутом огляду, кількістю зон, фокусною відстанню, геометрією кадру тощо – можна налаштувати безпосередньо через графічний інтерфейс, без потреби змінювати код.

Інтерфейс адаптований до практичних потреб користувача: достатньо вибрати відео, задати параметри камери та смуг, після чого натиснути “Run Detection” – і система виконає повний аналіз, відобразить швидкість над кожним транспортним засобом та дозволить оцінити ефективність візуально.

Така архітектура дозволяє швидко адаптувати систему під нові відео, експериментувати з параметрами зйомки, змінювати конфігурації та гнучко масштабувати функціональність, залишаючись при цьому зручною для кінцевого користувача.

Для підвищення зручності експлуатації системи розроблено графічний інтерфейс користувача (UI), який дозволяє конфігурувати параметри камери, сцени та візуалізації без потреби у взаємодії з кодом. Такий підхід забезпечує доступність інструменту навіть для не підготовлених користувачів та сприяє гнучкому налаштуванню системи під різні умови зйомки. На рисунку 3.1 наведено вікно основної конфігурації застосунку.

The screenshot shows a configuration window with the following fields and controls:

- Video Path:** speed.mp4 (with a Browse button)
- Video Preset:** Multi-Lane Highway (dropdown menu)
- Camera Parameters:**
 - Height (m): 8.0
 - Angle (deg): 40
 - Frame HFOV (deg): 70
 - Tilt (deg): 5
 - Yaw (deg): 0
 - Sensor Size (mm): 4.8
 - Focal Length (mm): 3.5
 - Num Zones: 10 (spinner)
 - Update Zones button
- Lane Configuration:**
 - Number of Lanes: 3 (spinner)
 - Lane 1 X-Range (start, end): 0, 340
 - Lane 2 X-Range (start, end): 340, 680
 - Lane 3 X-Range (start, end): 680, 1020
 - Lane 4 X-Range (start, end): (empty)
 - PPM Scale: 1.1, 1.0, 0.9, (empty)
- Speed Line Y:** 350 (spinner)
- Lens Distortion Correction:**
 - Apply Distortion Correction
 - fx: 1000, fy: 1000
 - cx: 510, cy: 250
 - Distortion Coeffs (k1,k2,p1,p2): 0,0,0,0
- Visualization:**
 - Show Zones
- Buttons:** Save Config, Load Config, Run Detection

Рисунок 3.1 – UI програми де відбувається конфігурація

Інтерфейс реалізовано у вигляді віконного застосунку на основі бібліотеки Tkinter. Основне вікно містить кілька функціональних блоків. У верхній частині користувач може обрати відеофайл для обробки, а також застосувати один із заздалегідь визначених пресетів сцени, наприклад "Multi-Lane Highway", який автоматично встановлює параметри камери та зони вимірювання.

Блок Camera Parameters дозволяє задавати фізичні параметри сцени: висоту камери, кут нахилу, кут поля зору (FOV), фокусну відстань та розмір сенсора. Також задається кількість зон (Num Zones), які використовуються для побудови перспективної шкали пікселів на метр (PPM). Кнопка "Update Zones" оновлює зони згідно з заданими параметрами.

У секції Lane Configuration користувач може вказати кількість смуг руху та координати меж кожної смуги по горизонталі (X-діапазон), а також задати індивідуальні коефіцієнти масштабування (PPM Scale) для кожної смуги. Це дозволяє враховувати варіації перспективи на різних ділянках зображення.

Параметр Speed Line Y відповідає за розташування контрольної лінії, яку об'єкти повинні перетнути для фіксації швидкості.

Блок Lens Distortion Correction дозволяє ввести параметри внутрішньої калібровки камери, зокрема фокусні відстані (f_x , f_y), координати оптичного центру (s_x , s_y) та коефіцієнти дисторсії (k_1 , k_2 , p_1 , p_2). За потреби користувач може увімкнути або вимкнути корекцію спотворень.

Секція Visualization дає змогу активувати графічне накладання зон PPM на відео. Внизу розташовані кнопки для збереження та завантаження конфігурацій, а також запуску детекції відео. У результаті після обробки відкривається вікно з відеопотоком, на якому у режимі реального часу відображаються прямокутники детектованих об'єктів, їхні ідентифікатори та швидкість у км/год.

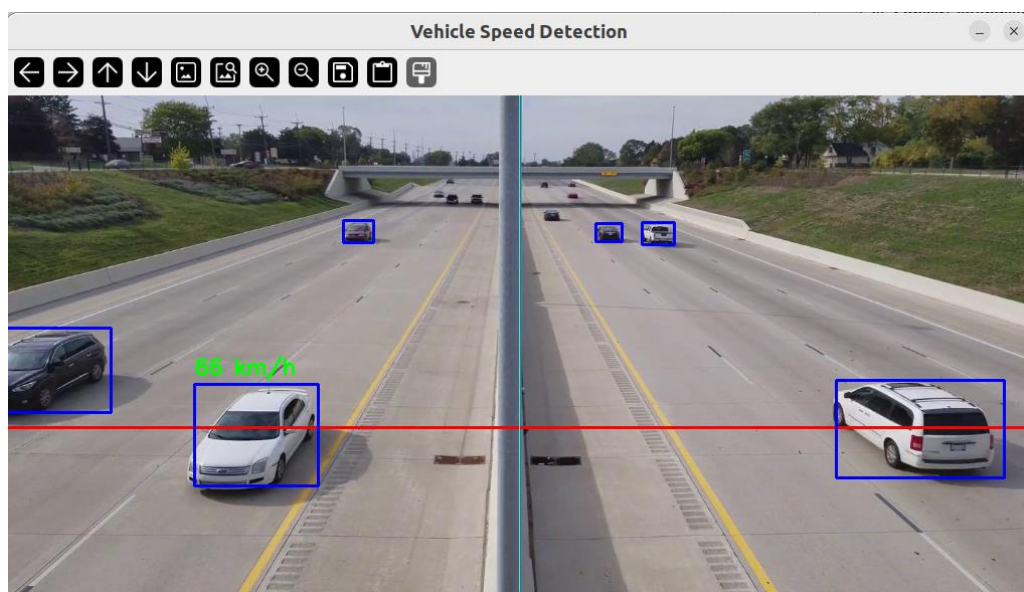


Рисунок 3.2 – Інтерфейс застосунку

Після запуску обробки відео користувач отримує інтерактивне вікно з візуалізацією результатів. На зображенні показано, як система автоматично виявляє транспортні засоби за допомогою донавченої моделі YOLOv10n. Кожен виявлений об'єкт обрамлено синім прямокутником, а також містить цифрову позначку швидкості (у км/год), якщо він перетнув контрольну червону лінію.

Контрольна лінія, розташована горизонтально на зображенні, слугує орієнтиром для моменту фіксації швидкості. Коли центр об'єкта перетинає цю лінію, система розраховує швидкість на основі пройденої піксельної відстані, масштабу (PPM) та часової різниці між кадрами. Наприклад, на зображенні видно автомобіль із визначеною швидкістю 66 км/год, що відображається зеленим шрифтом безпосередньо над об'єктом.

Крім цього, у верхній частині інтерфейсу розташовано панель керування з піктограмами, що дозволяють призупинити відтворення, зберегти кадр, змінити масштаб або здійснити перегляд у зручному режимі. Така візуалізація дозволяє не лише бачити результат детекції в реальному часі, але й оцінювати ефективність роботи трекінгу та точність накладання об'єктів на відео.

Завдяки цій функціональності система є повноцінним інструментом для моніторингу дорожнього руху та аналізу поведінки транспортних засобів у динаміці.

3.3 Аналіз результатів детекції об'єктів класу «автомобіль»

3.3.1 Просторовий та розмірний розподіл міток у навчальній вибірці

Для якісного аналізу навчального набору даних важливо оцінити не лише характеристики окремих bounding boxes, а й загальну структуру та щільність міток. Таке візуальне представлення дозволяє зрозуміти розподіл об'єктів за координатами, розмірами та кількістю, а також виявити потенційні перекоси у вибірці, які можуть вплинути на результати донавчання моделі.

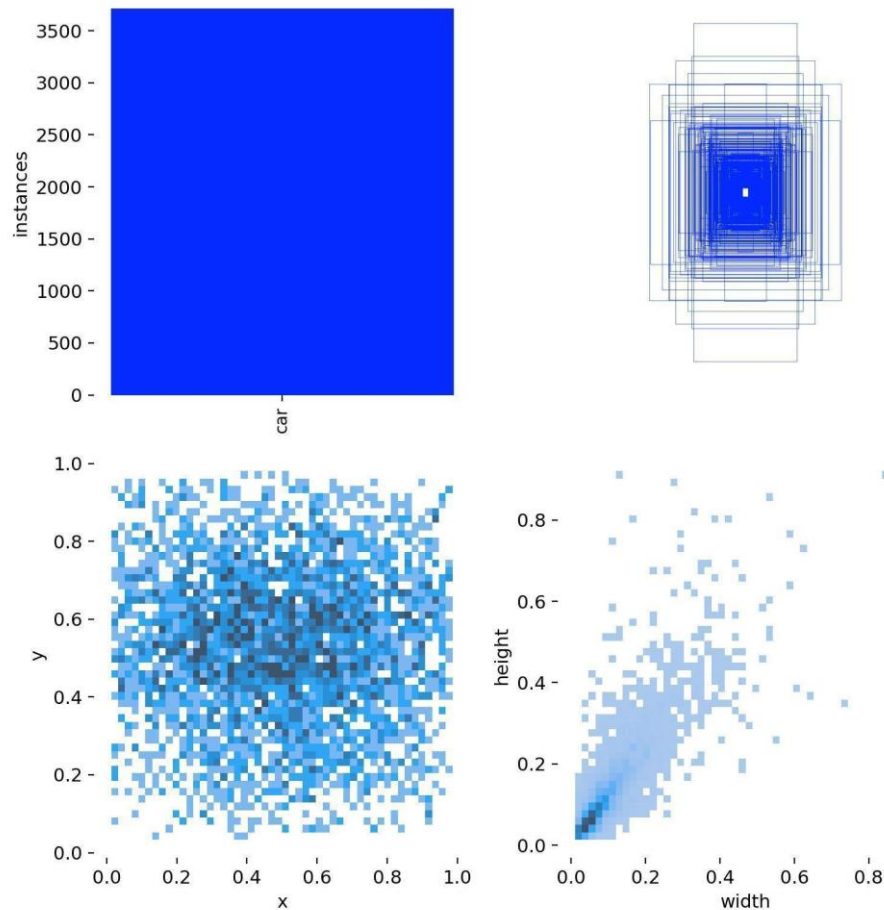


Рисунок 3.3 – Розподіл міток у навчальній вибірці

На рисунку 3.3 зображено комплексну візуалізацію розподілу міток у навчальній вибірці. Верхній лівий графік показує кількість інстансів класу «car», яка перевищує 3500 зразків – це свідчить про добре збалансовану кількість прикладів для цільового класу. У правій верхній частині наведено проєкцію bounding boxes у нормалізованих координатах (від 0 до 1), яка демонструє, що більшість рамок мають схожу геометрію і розташовані компактно в центрі зображення.

У нижньому лівому графіку показано двовимірний розподіл центрів bounding boxes за координатами (x, y). Центри концентруються навколо середини кадру, що відповідає типовому розміщенню транспортних засобів у полі зору дорожньої камери. Нижній правий графік відображає розподіл розмірів bounding boxes за висотою та шириною. Видно, що значна частина об'єктів має невеликі розміри – більшість знаходиться в межах ширини до 0.4 та висоти до 0.3 у

нормалізованих одиницях, що характерно для зйомки з відстані або з великої висоти.

Представлений графік дозволяє зробити висновок, що навчальна вибірка є збалансованою як за кількістю, так і за просторовими характеристиками об'єктів, що позитивно впливає на стабільність навчального процесу моделі YOLOv10n.

3.3.2 Результати тестування моделі YOLO

Для оцінки якості навчання моделі YOLOv10n були використані метрики, розраховані на навчальній та валідаційній вибірках. У таблиці 3.1 наведено значення precision, recall, F1-score та mAP@0.5, що характеризують ефективність виявлення об'єктів під час тренування. Ці показники дозволяють оцінити здатність моделі точно і повно ідентифікувати об'єкти класу car без перенавчання.

Таблиця 3.1 – Таблиця з метриками для навчальної та валідаційної вибірки

Тип вибірки	Клас	Precision	Recall	mAP@0.5	F1-score
Навчальна	car	0.917	0.845	0.939	0.879
Валідаційна	car	0.853	0.727	0.833	0.785

Отримані результати свідчать про високу якість навчання моделі. На навчальній вибірці precision становить 0.917, recall 0.845, а F1-score 0.879, що вказує на ефективне виявлення об'єктів без значного перенавчання. На валідаційній вибірці значення дещо нижчі: precision 0.853, recall 0.727, F1-score 0.785, що є очікуваним при переході до нових даних. Така стабільність метрик підтверджує надійність моделі для подальшого використання.

Після завершення навчання було проведено тестування моделі YOLOv10n на незалежному тестовому наборі даних, що включає понад 100 зображень отриманих з відеопотоку з дорожніми сценами, які були розмічені вручну за допомогою онлайн-платформи Roboflow. У процесі розмітки були вказані координати об'єктів класу «car» з урахуванням їхнього положення в кадрі та

масштабу. Такий набір забезпечив незалежність оцінювання від навчальних даних і дозволив об'єктивно протестувати точність виявлення транспортних засобів у реалістичних умовах.

Далі розглянемо графіки залежності precision, recall, F1-score від порогу впевненості, а також PR-криву, що демонструє взаємозв'язок між точністю і повнотою. Крім того, проаналізуємо матрицю помилок (confusion matrix), що відображає частку правильних і помилкових класифікацій [38].

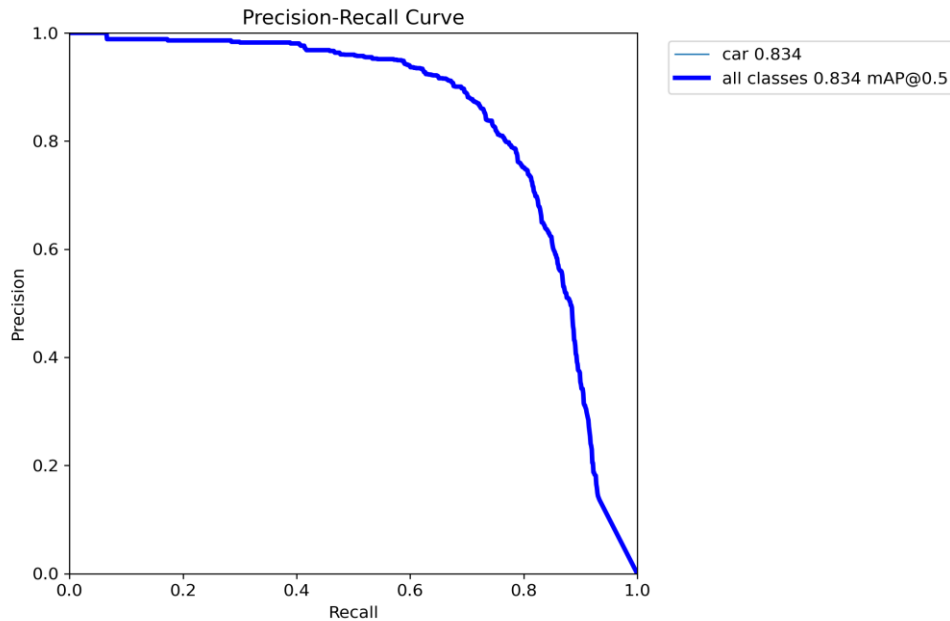


Рисунок 3.4 – Precision-Recall крива

На рисунку 3.4 зображено Precision-Recall криву, яка демонструє зміну точності (Precision) залежно від повноти (Recall) для моделі YOLOv10n, донавченої на спеціалізованому датасеті. Крива побудована на основі передбачень лише для одного класу – «car».

Як видно з графіка, при низьких значеннях Recall (менше 0.3) точність знаходиться практично на рівні 1.0, що свідчить про високу достовірність виявлених об'єктів. У міру зростання Recall крива поступово знижується, демонструючи класичний компроміс між кількістю виявлених об'єктів і якістю цих виявлень.

Інтегральне значення середньої точності (mAP@0.5) становить 0.834, що є дуже хорошим результатом і свідчить про здатність моделі точно виявляти та

класифікувати об'єкти. Це підтверджує високу якість адаптації моделі до поставленої задачі та відповідність отриманих результатів очікуваному рівню продуктивності.

Крива F1-score відображає гармонійне середнє між метриками точності (Precision) та повноти (Recall) при різних значеннях порогу впевненості моделі. Такий графік є корисним для визначення оптимального балансу між виявленням об'єктів і мінімізацією помилкових класифікацій.

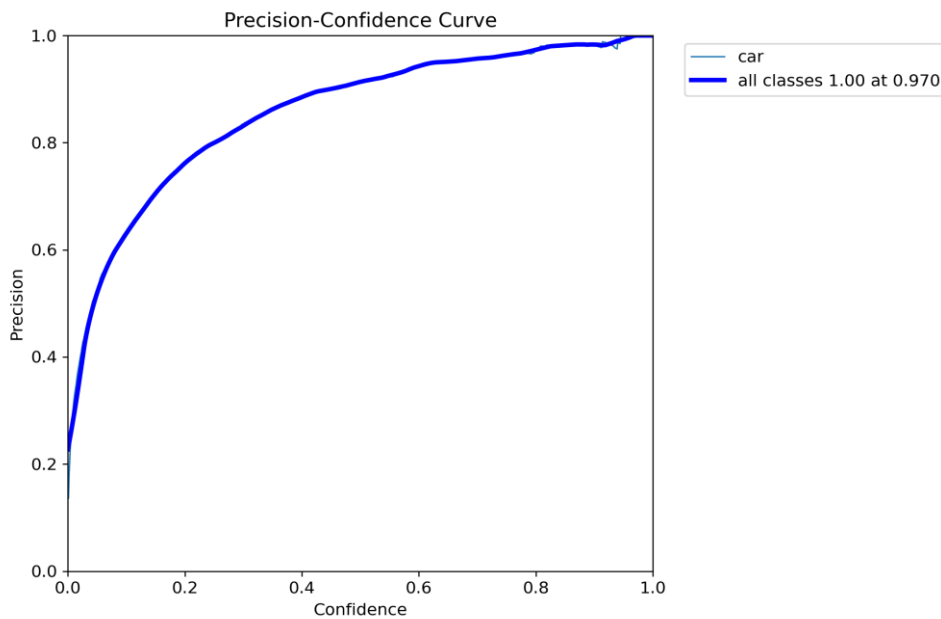


Рисунок 3.5 – Крива точності (Precision)

Графік на рисунку 3.5 ілюструє зміну метрики точності (Precision) залежно від порогу впевненості моделі при класифікації об'єктів як автомобілі. Precision відображає, яка частка виявлених об'єктів дійсно належить до класу «car», тобто дозволяє оцінити рівень хибно позитивних спрацювань (false positives).

Крива має чіткий зростаючий характер: при низьких значеннях confidence (менше 0.2) точність становить близько 20%, що свідчить про значну кількість помилкових спрацювань. Проте із підвищенням порогу впевненості точність зростає, досягаючи максимальної оцінки 1.00 при пороговому значенні 0.97. Це означає, що за умови високої впевненості модель практично не припускається помилок і виявляє лише справжні об'єкти.

Такий графік дає змогу підібрати оптимальне значення порогу впевненості залежно від вимог до точності виявлення. Наприклад, за високої важливості мінімізації хибнопозитивних результатів поріг встановлюється ближче до 0.9 або вище, що супроводжується зменшенням повноти (Recall). Крива Precision-Confidence відображає взаємозв'язок між жорсткістю та гнучкістю детекції, дозволяючи керувати компромісом між ними.

Крім точності (Precision), важливо також аналізувати здатність моделі виявляти всі релевантні об'єкти – тобто оцінювати повноту (Recall). Крива Recall-Confidence демонструє, як змінюється цей показник залежно від встановленого порогу впевненості класифікації.

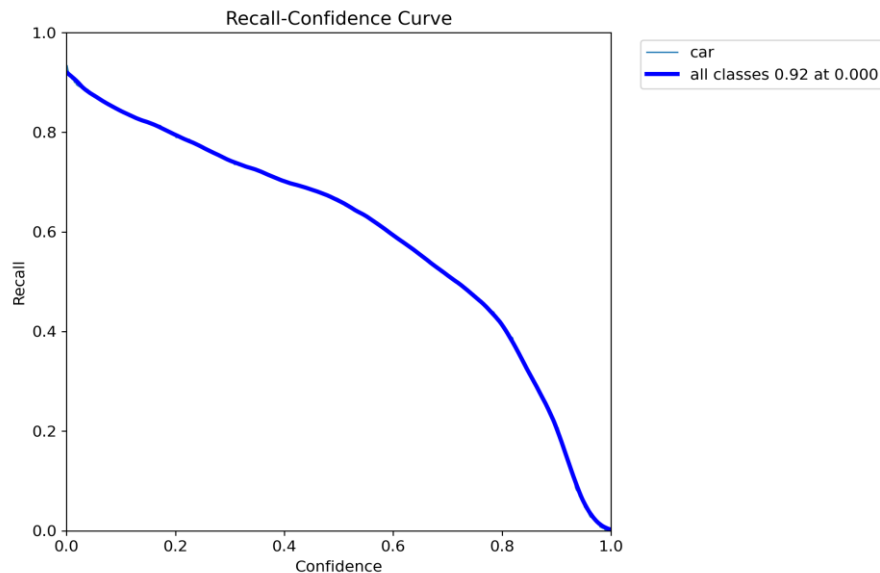


Рисунок 3.6 – Крива повноти (Recall)

На рисунку 3.6 зображено криву Recall-Confidence для класу «car». Вона показує, що при дуже низькому порозі впевненості (confidence ≈ 0.0) модель здатна виявляти до 92% реальних об'єктів у кадрі, що відповідає максимально можливому значенню повноти. Це відбувається за рахунок того, що модель допускає більшу кількість менш впевнених передбачень.

Однак зі збільшенням порогу впевненості значення recall поступово знижується. Наприклад, при confidence > 0.9 повнота падає нижче 20%, що свідчить про втрату значної частини цільових об'єктів – модель стає «надто обережною» і відкидає багато справжніх детекцій.

Такий результат є типовим прикладом компромісу між Recall та Precision: для максимальної повноти потрібно знижувати поріг, ризикуючи при цьому збільшити кількість false positives. Крива дає змогу підібрати поріг, що забезпечує потрібний рівень чутливості моделі до виявлення автомобілів.

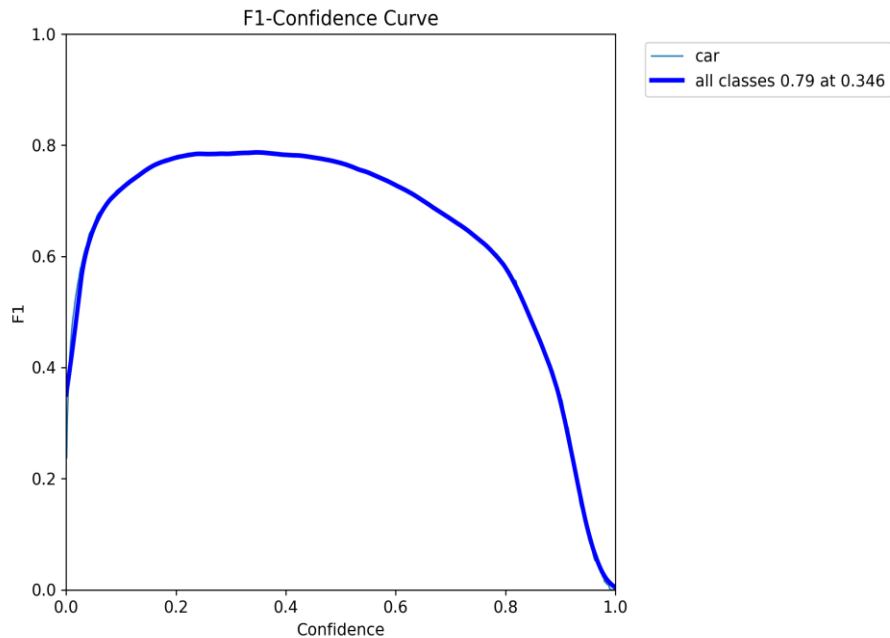


Рисунок 3.7 – Крива F1-score

Щоб збалансувати обидва підходи – точність (Precision) та повноту (Recall) – застосовується F1-score. Це гармонійне середнє між Precision і Recall, яке дозволяє отримати інтегральну оцінку загальної якості моделі.

На рисунку 3.7 зображено, як змінюється значення F1-score залежно від порогу confidence. Спостерігається типовий для якісної моделі тренд: початкове зростання F1 при збільшенні впевненості до певного оптимального рівня, після чого – поступове зниження.

У рамках проведеного дослідження максимальне значення F1-метрики становить 0.79 при пороговому значенні впевненості 0.346. Це означає, що саме за цього порогу досягається найкращий баланс між точністю та повнотою, можна використовувати в реальній експлуатації моделі для забезпечення максимальної збалансованості результатів.

Оцінювання якості класифікації моделі також передбачає аналіз типів помилок, які вона допускає. Для цього будується матриця помилок (confusion

matrix), яка демонструє співвідношення між фактичними та передбаченими класами. Це дозволяє виявити, скільки об'єктів модель правильно класифікувала, а скільки – помилково.

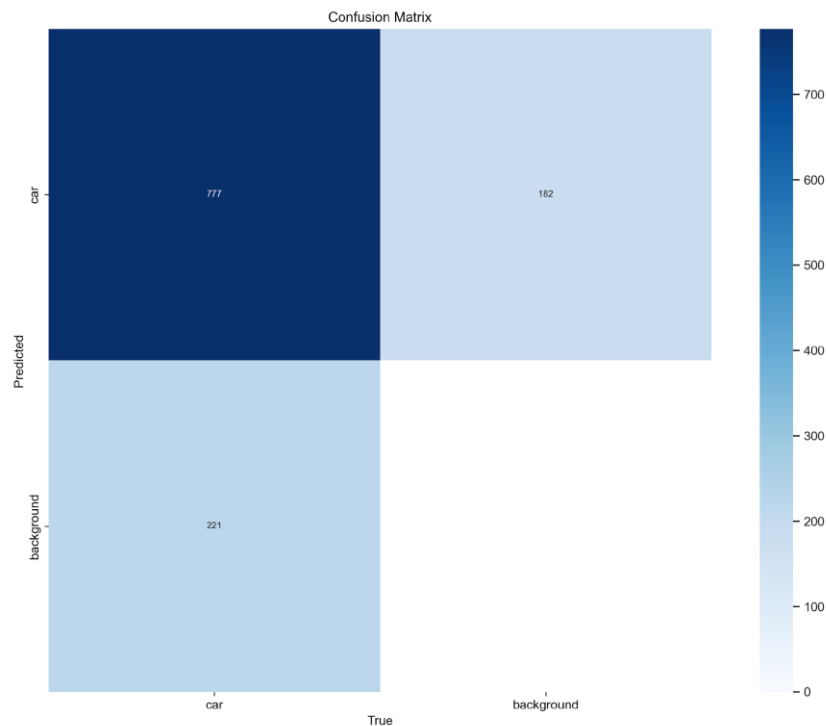


Рисунок 3.8 – Матриця помилок

На рисунку 3.8 подано матрицю помилок для двох класів: «car» (автомобіль) і «background» (фон). Верхній лівий квадрат містить значення 777 – кількість випадків, коли модель правильно виявила автомобілі (true positives). У правому верхньому квадраті знаходиться 152 – це хибнопозитивні передбачення, коли модель помилково класифікувала фонові області як автомобілі.

Нижній лівий квадрат показує 231 хибнонегативних результатів – це випадки, коли реальні автомобілі були проігноровані моделлю та класифіковані як фон. Нижній правий квадрат (true negatives) у цьому контексті не має змістового навантаження, оскільки не враховується у виведенні класу «background» як об'єкта.

Такий розподіл помилок демонструє загальну тенденцію: модель досить точно розпізнає автомобілі, однак ще має простір для зменшення кількості false negatives (пропущених автомобілів). Аналіз цієї матриці дозволяє точніше скоригувати параметри моделі та поріг confidence для покращення результатів.

Для точнішої інтерпретації результатів класифікації, особливо у випадках із нерівномірним розподілом об'єктів за класами, використана нормалізована матриця помилок. Вона відображає частки правильних і помилкових класифікацій у межах кожного класу, незалежно від абсолютної кількості зразків.

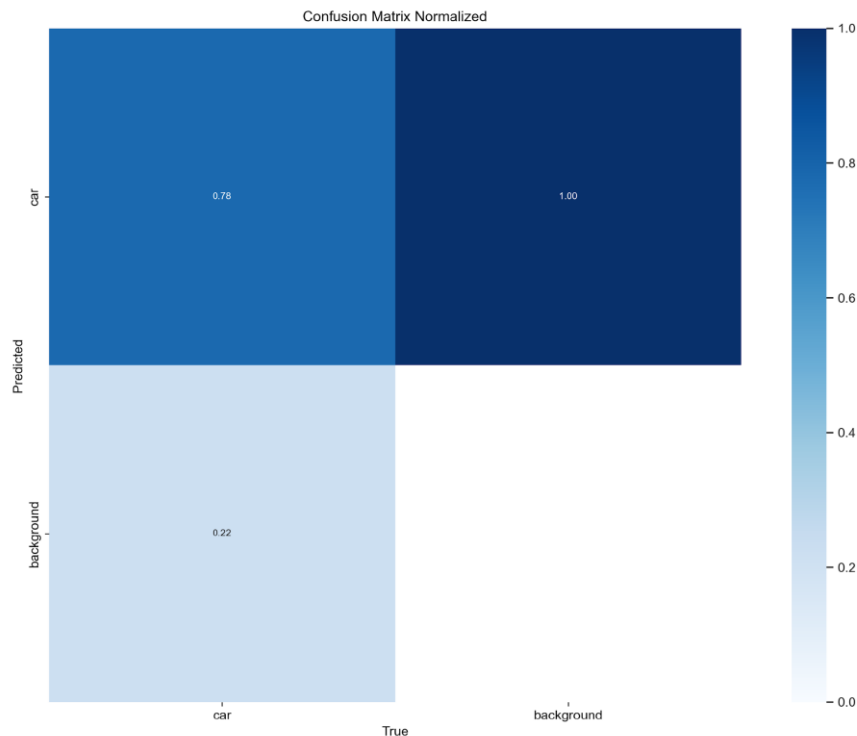


Рисунок 3.9 – Нормалізована матриця помилок

На рисунку 3.9 представлено нормалізовану матрицю помилок для двох класів: «car» (автомобіль) та «background» (фон). З аналізу матриці видно, що 78 % об'єктів класу «car» були правильно класифіковані як автомобілі, тобто спрацювали як true positives. Водночас 22 % автомобілів були неправильно віднесені до фону, що свідчить про наявність хибнонегативних результатів (false negatives). Об'єкти фону, у свою чергу, були класифіковані коректно у 100 % випадків, що означає повну відсутність хибнопозитивних спрацювань моделі при класифікації класу «background».

Це свідчить про високу специфічність моделі щодо фону, але також вказує на потребу покращити recall – здатність моделі не пропускати об'єкти класу «car». Нормалізація дозволяє легко порівнювати ефективність моделі на різних вибірках, навіть за наявності сильного класового дисбалансу [39].

Результати оцінки моделі YOLOv10n на незалежному тестовому наборі даних, подані в таблиці 3.2

Таблиця 3.2 – Таблиця з метриками для тестувальної вибірки

Клас	Precision	Recall	mAP@0.5	F1-score
car	0.86	0.73	0.834	0.79

Як видно з таблиці, модель демонструє високу точність при виявленні об'єктів класу «car» (Precision = 0.86), водночас підтримуючи задовільний рівень повноти (Recall = 0.73). Максимальне значення F1-метрики досягається при порозі впевненості 0.346, становлячи 0.79, що свідчить про збалансовану роботу моделі. Середнє значення mAP@0.5 = 0.834 підтверджує загальну ефективність моделі в задачі детекції автомобілів.

3.4 Оцінка точності визначення швидкості транспортних засобів

Основною метою запропонованого метода є точне визначення швидкості руху транспортних засобів на основі відеоінформації. Для оцінки ефективності реалізованого методу необхідно здійснити перевірку його достовірності шляхом порівняння з еталонними (істинними) значеннями.

У цьому дослідженні використано ручний підхід до визначення істинної швидкості: обирається контрольний відрізок дороги з відомою довжиною (у метрах), а час проходження цього відрізка транспортним засобом визначається за кількістю кадрів на відео та частотою відеозапису (FPS) [40].

Таким чином, для кожного транспортного засобу можна визначити реальну швидкість та порівняти її з тією, яку прогнозує метод. Це дозволяє обчислити абсолютну та відносну похибки, а також візуалізувати ефективність запропонованого методу за допомогою графіків.

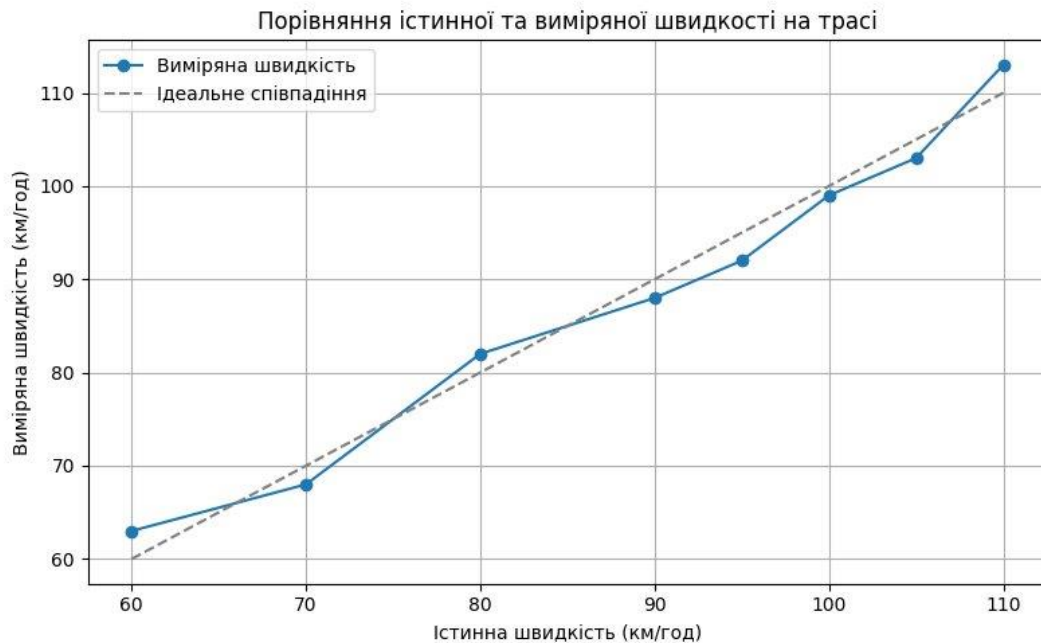


Рисунок 3.10 – Порівняння істинної та вимірної швидкості на трасі

На рисунку 3.10 зображено результати експериментального тестування методу визначення швидкості транспортних засобів на реальному відеофрагменті дорожнього руху. Для перевірки точності було обрано ділянку автомагістралі з відомою протяжністю контрольного відрізка (50 метрів), по якому протягом запису проїжджали автомобілі. Істинна швидкість визначалась вручну – за допомогою вимірювання кількості кадрів, які потрібні автомобілю, щоб подолати цю дистанцію, з урахуванням частоти відеозапису (FPS = 30).

Отримані значення були використані як еталонні (істинні швидкості), після чого результати порівнювались із вимірними швидкостями, які прогнозував метод. Синя лінія на графіку відображає вимірні значення, тоді як пунктирна сіра – ідеальне співпадіння з істинними швидкостями.

Аналіз графіка демонструє високу кореляцію між розрахованими та реальними швидкостями. В усіх протестованих випадках відхилення не перевищувало $\pm 5\%$, що відповідає очікуваній похибці для систем на основі комп'ютерного зору. Такий рівень точності підтверджує практичну придатність запропонованого підходу для застосування в умовах реального дорожнього середовища, зокрема на швидкісних трасах.

Висновки до розділу 3

У розділі описана реалізація методу визначення швидкості руху транспортних засобів у відеопотоці з використанням нейромережових засобів комп'ютерного зору. Метод передбачає поєднання етапів детекції об'єктів, їхнього відстеження між послідовними кадрами, розрахунку зміщення у пікселях, масштабування в метричні одиниці за допомогою коефіцієнта PPM, та подальшого визначення швидкості з урахуванням частоти відеопотоку. Для зменшення флуктуацій у розрахунках було реалізовано згладжування значень швидкості.

З метою перевірки працездатності методу розроблено програмну реалізацію, що дозволяє конфігурувати параметри сцени, запускати аналіз відео та візуалізувати результати. Метод базується на донавченій моделі YOLOv10n, адаптованій до задачі виявлення автомобілів, і трекері DeepSORT для побудови траєкторій об'єктів у кадрі.

Проведене тестування підтвердило ефективність реалізованого підходу: точність (precision) детекції досягла 0.86, повнота (recall) — 0.73, F1-score — 0.79 при порозі 0.346, а середнє значення mAP@0.5 перевищило 0.83. Аналіз нормалізованої матриці неточностей показав надійне розрізнення класу «автомобіль» та фону. Похибка оцінки швидкості в реальних умовах, визначена експериментально на контрольній ділянці дороги, не перевищувала 5%, що свідчить про високу точність запропонованого методу.

Таким чином, реалізований метод відповідає поставленим у роботі завданням та може бути основою для подальшого створення повноцінних систем автоматичного відеоконтролю швидкісного режиму в умовах реального трафіку.

Загальні висновки

Метою кваліфікаційної роботи бакалавра було підвищення точності визначення швидкості руху автомобілів у відеопотоці засобами глибокого навчання.

Для досягнення мети були поставлені та виконані такі задачі:

- провести аналіз існуючих методів комп'ютерного зору та глибокого навчання для виявлення та трекінгу транспортних засобів у відеопотоці;
- розроблено метод визначення швидкості руху виявлених транспортних засобів у відеопотоці на основі координат і часових міток;
- підібрано та підготовлено відомий набір даних із зображень автомобілів, що включає різні умови зйомки;
- створено програму реалізацію методу з підтримкою детекції, трекінгу, розрахунку швидкості та візуалізації результатів;
- проведено експерименти для оцінки точності запропонованого методу визначення швидкості руху автомобіля.

Розроблений метод поєднує в собі сучасні підходи до детекції об'єктів, зокрема модель YOLOv10n, адаптовану до умов дорожнього трафіку, та трекінг на основі DeepSORT, що забезпечує побудову траєкторій об'єктів.

Для тестування методу створено програмну реалізацію, що забезпечує візуалізацію результатів і дозволяє налаштовувати параметри аналізу.

Проведений аналіз точності моделі підтвердив коректність обраного підходу до виявлення та трекінгу транспортних засобів, а також розрахунку їх швидкості у відеопотоці. Значення показників якості – $\text{precision} = 0.86$, $\text{recall} = 0.73$, $\text{mAP}@0.5 > 0.83$ – демонструють високу точність класифікації та виявлення автомобілів, а нормалізована матриця помилок засвідчила добру відокремлюваність фону.

Метод може бути інтегрований у практичні системи моніторингу та контролю швидкості дорожнього руху, зокрема завдяки можливості працювати у режимі, близькому до реального часу.

Перелік посилань

1. Lian H., Li M., Li T., Zhang Y., Shi Y., Fan Y., Yang W., Jiang H., Zhou P., Wu H. Vehicle speed measurement method using monocular cameras. *Scientific Reports*. Vol. 15, 2025. URL: <https://doi.org/10.1038/s41598-025-87077-6>
2. Ultralytics YOLO Docs. URL: <https://docs.ultralytics.com/>
3. DeepSORT. Mastering Deep Sort: The Future of Object Tracking Explained. URL: <https://www.ikomia.ai/blog/deep-sort-object-tracking-guide>
4. CNN. What are convolutional neural networks? URL: <https://www.ibm.com/think/topics/convolutional-neural-networks>
5. Optical flow. Computational Neuroscience. URL: <https://soniajoseph.github.io/computational-neuroscience/>
6. GoodFeaturesToTrack. OpenCV projects – How to extract features from the image in Python? URL: <https://datahacker.rs/004-opencv-projects-how-to-extract-features-from-the-image-in-python/>
7. Camlytics Vehicle Speed Detection. Vehicle speed detection solution. URL: <https://camlytics.com/solutions/calculate-vehicle-speed>
8. Sangsuwan K., Ekpanyapong M. Video-Based Vehicle Speed Estimation Using Speed Measurement Metrics. *IEEE Access*. 2024. Vol. 12. P. 4845–4858. URL: <https://doi.org/10.1109/ACCESS.2024.3350381>.
9. Real-time Vehicle Detection and Speed Estimation System / B. Nikumbhe et al. *International Journal of Scientific Research in Engineering and Management*. 2022. Vol. 6. P. 1–6. URL: https://www.researchgate.net/publication/373952515_Real-time_Vehicle_Detection_and_Speed_Estimation_System
10. Real-time Vehicle Detection and Speed Estimation System. Real-Time Vehicle Detection, Tracking, Number Plate Detection, and Speed Estimation using Raspberry Pi HQ Camera. URL: <https://community.element14.com/challenges-projects/project14/photography/b/blog/posts/vehicle-x-real-time-vehicle-detection-tracking-number-plate-detection-and-speed-estimation-using-raspberry-pi-hq-camera>

11. Macko A., Gajdošech L., Kocur V. Efficient Vision-based Vehicle Speed Estimation. URL: <https://arxiv.org/abs/2505.01203>.
12. Enhanced YOLOv5s + DeepSORT method for highway vehicle speed detection and multi-sensor verification / Z. Luo et al. *Frontiers in Physics*. 2024. Volume 12, 2024. URL: <https://doi.org/10.3389/fphy.2024.1371320>.
13. Soybean–Corn Seedling Crop Row Detection for Agricultural Autonomous Navigation Based on GD-YOLOv10n-Seg / T. Sun et al. *Agriculture*. 2025. Vol. 15, no. 7. URL: <https://doi.org/10.3390/agriculture15070796>.
14. Boesch G. YOLOv10: Real-Time Object Detection Evolved. *viso*. 2024. URL: https://viso.ai/deep-learning/yolov10/?utm_source=chatgpt.com.
15. Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once / V. Adibhatla et al. *Mathematical Biosciences and Engineering*. 2021. Vol. 18. P. 4411–4428. URL: <https://doi.org/10.3934/mbe.2021223>.
16. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon et al. URL: <https://arxiv.org/abs/1506.02640>.
17. Kenk, M. Detection in Adverse Weather Nature (DAWN) dataset (Version 3) [Data set]. Mendeley Data. URL: <https://data.mendeley.com/datasets/766ygrbt8y/3>
18. Roboflow. Everything you need to build and deploy computer vision applications. URL: <https://roboflow.com/>
19. Ultralytics YOLO Docs. Object Detection Datasets Overview (Ultralytics YOLO format – train/val/test). URL: <https://docs.ultralytics.com/datasets/detect/>
20. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. URL: <https://arxiv.org/abs/1412.6980>.
21. Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data*. 2019, Vol.6. URL: <https://doi.org/10.1186/s40537-019-0197-0>
22. Ultralytics YOLO Docs. Train Mode – Hardware Requirements and GPU Setup. 2024. URL: <https://docs.ultralytics.com/modes/train/#hardware-requirements>
23. Roboflow. Transfer Learning with YOLOv10 on Custom Object Classes. Roboflow Blog, 2024. URL: <https://blog.roboflow.com/yolov10-how-to-train/>

24. Ultralytics YOLO Docs. Loss Functions in YOLO (objectness, classification, bbox regression). 2024. URL: <https://docs.ultralytics.com/usage/losses/>
25. Fernández Llorca D., Hernández Martínez A., García Daza I. Vision-based vehicle speed estimation: A survey. *IET Intelligent Transport Systems*. 2021. Vol. 15, no. 8, p. 987–1005. URL: <https://arxiv.org/abs/2101.06159>
26. Zhang M., Lu J. Object Detection and Bounding Boxes. Dive into Deep Learning. 2023. URL: https://d2l.ai/chapter_computer-vision/bounding-box.html
27. Wojke N., Bewley A., Paulus D. Simple Online and Realtime Tracking with a Deep Association Metric. URL: <https://arxiv.org/abs/1703.07402>.
28. Setiyono B., Sulistyanningrum D.R., Soetrisno, Wicaksono D.W. Multi Vehicle Speed Detection Using Euclidean Distance Based on Video Processing. *International Journal of Computing*. 2019. URL: <https://doi.org/10.47839/ijc.18.4.1613>
29. Redondo E. L., Martinez-Marin T. A compact representation of the environment and its frontiers for autonomous vehicle navigation. 2016 IEEE Intelligent Vehicles Symposium (IV). 2016. P. 851–857. URL: <https://doi.org/10.1109/IVS.2016.7535487>.
30. Sochor J., Juránek R., Herout A. Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding*. 2017. Vol. 161. P. 87–98. URL: <https://doi.org/10.1016/j.cviu.2017.05.015>
31. Zhao, J., Yang, X., & Zhang, C. (2023). Vehicle trajectory reconstruction for intersections: an integrated wavelet transform and Savitzky-Golay filter approach. *Transportmetrica A: Transport Science*, 2024. <https://doi.org/10.1080/23249935.2022.2163207>
32. Torovets T. ROC-криві. Оглядова стаття. DOU.ua, 17 червня 2021. URL: <https://dou.ua/forums/topic/33858/>
33. TrainInData Precision-Recall Blog. Precision Recall Curves — Noor UI Huda, 17 Sep 2024. URL: <https://www.blog.trainindata.com/precision-recall-curves/>
34. Python Documentation. URL: <https://www.python.org/doc/>

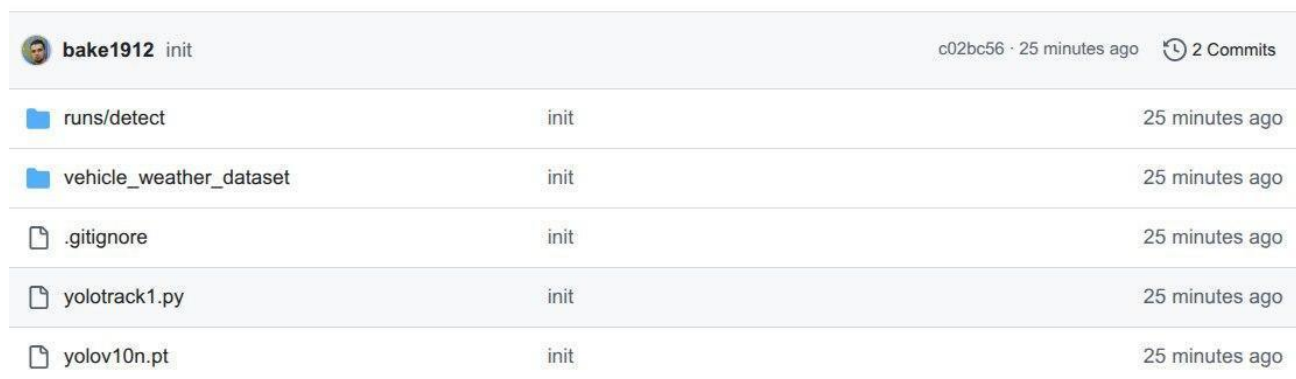
35. Harris C.R., Millman K.J., van der Walt S.J., et al. Array programming with NumPy. *Nature*, 2020, Vol. 585. URL: <https://www.nature.com/articles/s41586-020-2649-2>
36. Beniz D.B., Espíndola A.M. Using Tkinter of Python to create Graphical User Interface (GUI) for scripts in LNLS. Proceedings of PCaPAC 2016, *Campinas (Brazil)*, 2016. URL: <https://accelconf.web.cern.ch/pcapac2016/papers/wepoprpo25.pdf>
37. Python Library Docs. URL: <https://docs.python.org/3/library/json.html>
38. Saito T., Rehmsmeier M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets // *PLOS ONE*. 2015, Vol. 10, no. 3. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118432>
39. Phan T. H., Yamamoto K. Resolving Class Imbalance in Object Detection with Weighted Cross Entropy Losses. URL: <https://arxiv.org/abs/2006.01413>.
40. Epstein B., Westlake B. G. Determination of vehicle speed from recorded video using reverse projection photogrammetry and file metadata. *Journal of Forensic Sciences*, Vol. 64, 2019. URL: <https://doi.org/10.1111/1556-4029.14053>
41. Range Gowda S., Wolf P. A Dependable Trajectory Prediction Pipeline for Autonomous Driving: Integrating Computer Vision and Bayesian Networks for Highway Safety. 2025. URL: https://www.researchgate.net/figure/YOLO-classification-with-NMS-suppression_fig1_389418263
42. Lingineni, D., Dusi, P., Jakkam, R.S., Yada, S. (2023). Object Detection and Tracking Using DeepSORT. In: Buyya, R., Hernandez, S.M., Kovvur, R.M.R., Sarma, T.H. (eds) Computational Intelligence and Data Analytics. *Lecture Notes on Data Engineering and Communications Technologies*, vol 142. Springer, Singapore. URL: https://doi.org/10.1007/978-981-19-3391-2_23

Додатки

Додаток А

Програмний код

Посилання на репозиторій з програмним кодом розробленого методу:
<https://github.com/bake1912/KRB>



File	Commit Message	Time
runs/detect	init	25 minutes ago
vehicle_weather_dataset	init	25 minutes ago
.gitignore	init	25 minutes ago
yolotrack1.py	init	25 minutes ago
yolov10n.pt	init	25 minutes ago

Репозиторій містить папку `vehicle_weather_dataset` з датасетом (налаштування в `data.yaml`, зображення в `images/` та YOLO-мітки в `labels/`), файл моделі `yolov10n.pt` для детекції автомобілів, а також головний скрипт `yolotrack1.py`, що реалізує повний пайплайн: через GUI на `tkinter` дозволяє задати параметри камери та смуг руху, обчислює зони кадру для перерахунку пікселів у метри, запускає модель YOLO для детекції й трекінгу авто, оцінює швидкість при перетині заданої лінії, згладжує результати й виводить їх поверх відео в реальному часі.

Додаток Б

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Метод визначення швидкості руху автомобіля у
відеопотоці нейромеревими засобами

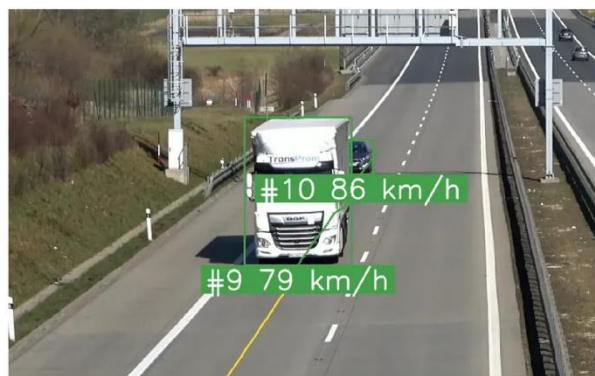
Виконав: Студент групи КН-21-1 Мельник Микола

Керівник: Доцент кафедри КН Багрій Руслан

2

Актуальність

Актуальність розробки зумовлена потребою у точному та автоматизованому контролі швидкості транспортних засобів у реальному часі для підвищення безпеки дорожнього руху.



Мета та завдання

Метою кваліфікаційної роботи бакалавра є підвищення точності визначення швидкості руху автомобіля у відеопотоці засобами глибокого навчання.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів комп'ютерного зору та глибокого навчання для виявлення та трекінгу транспортних засобів у відеопотоці;
- розробити метод визначення швидкості руху виявлених транспортних засобів у відеопотоці на основі координат і часових міток;
- підготувати набір даних, що містить різноманітні відеозаписи з різними умовами зйомки руху транспортних засобів.
- створити програму реалізацію методу з підтримкою детекції, трекінгу, розрахунку швидкості та візуалізації результатів;
- провести експерименти для оцінки точності запропонованого методу визначення швидкості руху автомобіля.

Загальна схема методу визначення швидкості руху автомобіля у відеопотоці

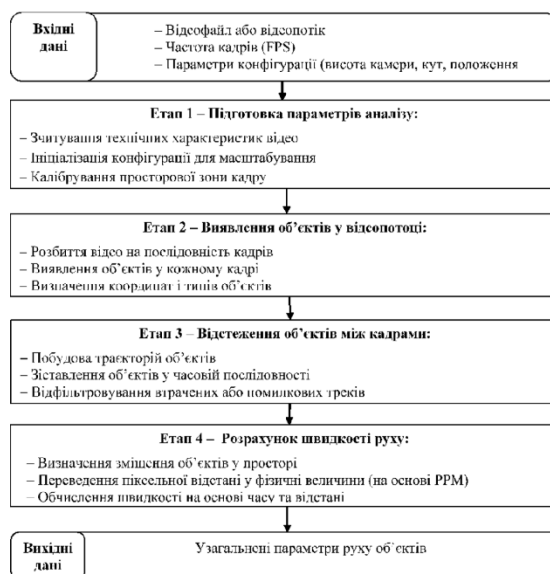
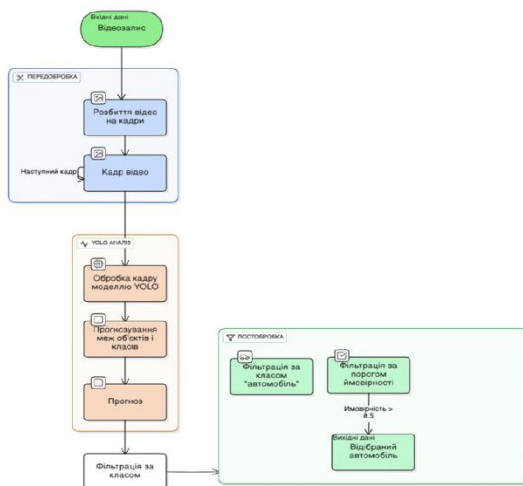
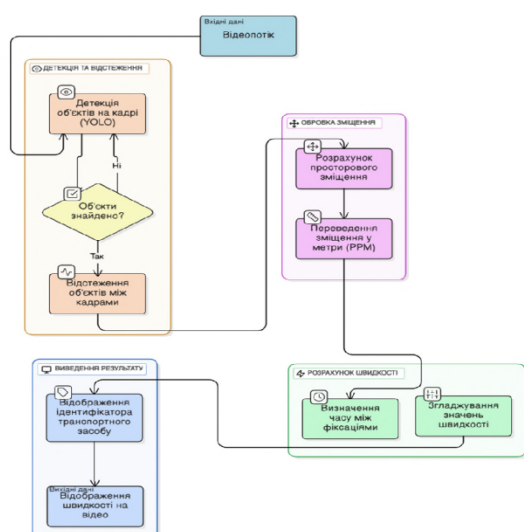


Схема виявлення автомобілів у відеопотоці за допомогою неймережі YOLO



1. Відео розбивається на кадри.
2. Кожен кадр обробляється неймережею YOLO, яка прогнозує межі об'єктів і їх класи.
3. Відфільтровуються лише об'єкти класу «автомобіль» з імовірністю понад 0.5.

Схема обчислення швидкості автомобілів у відеопотоці за допомогою неймережі YOLO



1. YOLO виявляє об'єкти в кожному кадрі, а DeepSORT відстежує їх між кадрами.
2. Розраховується піксельне зміщення об'єкта та переводиться в метри за допомогою коефіцієнта PPM.
3. Визначається час між кадрами та обчислюється швидкість. Дані згладжуються для стабільності.
4. Відео доповнюється швидкістю об'єкта на екрані.

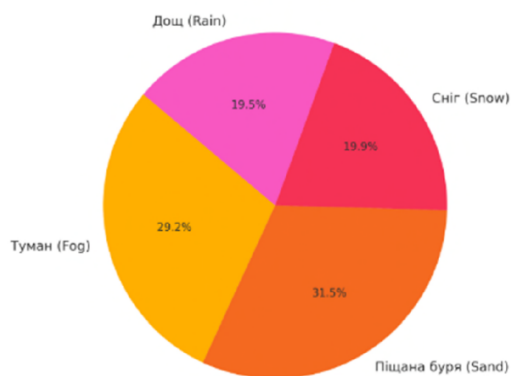
Опис обраного датасету

DAWN (Diverse Adverse Weather Needs) — це спеціалізований датасет для задач комп'ютерного зору, що містить відеозаписи дорожнього руху з камер спостереження та відкритих джерел.

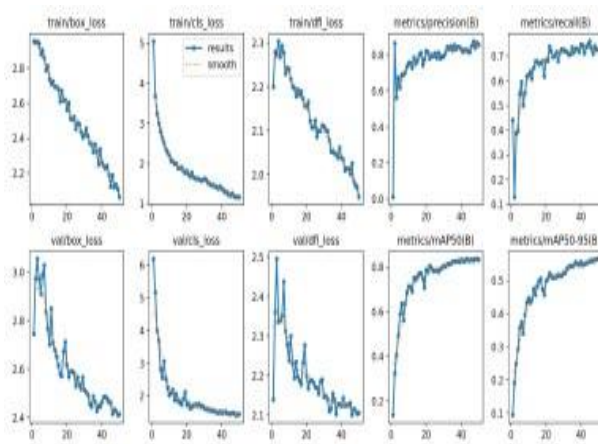
Він охоплює широкий спектр умов навколишнього середовища, включаючи туман, дощ, сніг та піщану бурю, і забезпечує високоточні анотації транспортних засобів.

Датасет включає різні умови освітлення (день, ніч, тіні) та ракурси зйомки (фронтальні, діагональні, з висоти). Датасет поділено на навчальну (80 %) та валідаційну (20 %) вибірки

Розподіл зображень у датасеті DAWN за погодними умовами



Оцінка якості моделі під час навчання



Тип вибірки	Клас	Precision	Recall	mAP@0.5	F1-score
Навчальна	car	0.917	0.845	0.939	0.879
Валідаційна	car	0.853	0.727	0.833	0.785

На навчальній вибірці precision становить 0.917, recall 0.845, а F1-score 0.879, що вказує на ефективне виявлення об'єктів без значного перенавчання.

На валідаційній вибірці значення дещо нижчі: precision 0.853, recall 0.727, F1-score 0.785, що є очікуваним при переході до нових даних.

Тестування на незалежному наборі даних

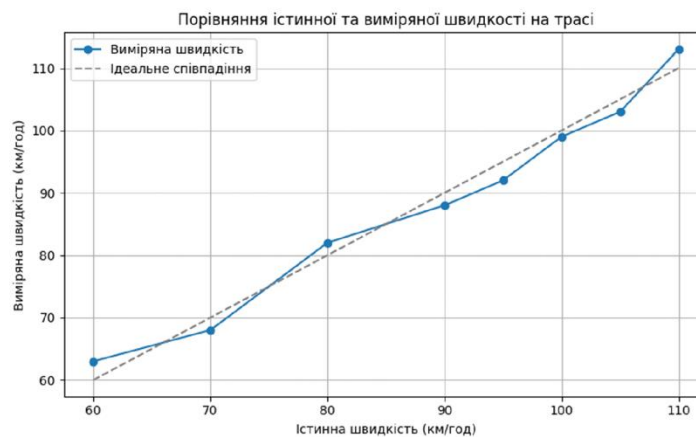
Клас	Precision	Recall	mAP@0.5	F1-score
car	0.86	0.73	0.834	0.79

Модель демонструє високу точність при виявленні об'єктів класу «car» (Precision = 0.86), водночас підтримуючи задовільний рівень повноти (Recall = 0.73).

Максимальне значення F1-метрики досягається при порозі впевненості 0.346, становлячи 0.79, що свідчить про збалансовану роботу моделі.

Середнє значення $mAP@0.5 = 0.834$ підтверджує загальну ефективність моделі в задачі детекції автомобілів.

Порівняння істинної та вимірної швидкості на трасі



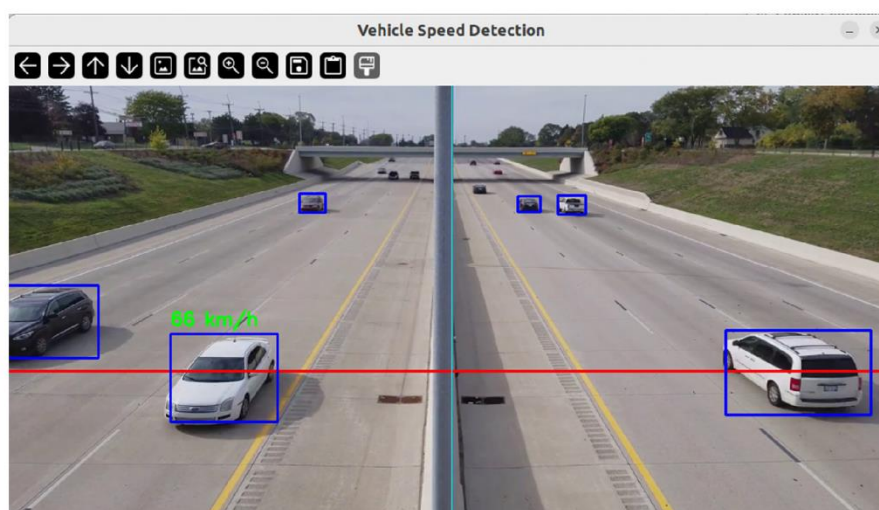
Конфігурація

The screenshot shows the 'Vehicle Speed Detection Configurator' window. It is divided into several sections:

- Video Path:** speed.mp4 (with a 'Browse' button)
- Video Preset:** Multi-Lane Highway
- Camera Parameters:**
 - Height (m): 8.0
 - Angle (deg): 40
 - Frame HFOV (deg): 70
 - Tilt (deg): 5
 - Yaw (deg): 0
 - Sensor Size (mm): 4.8
 - Focal Length (mm): 3.5
 - Num Zones: 10
 - Update Zones button
- Lane Configuration:**
 - Number of Lanes: 3
 - Lane 1 X-Range (start, end): 0, 340, PPM Scale: 1.1
 - Lane 2 X-Range (start, end): 340, 680, PPM Scale: 1.0
 - Lane 3 X-Range (start, end): 680, 1020, PPM Scale: 0.9
 - Speed Line Y: 350
- Lens Distortion Correction:**
 - Apply Distortion Correction
 - fx: 1000, fy: 1000
 - cx: 510, cy: 250
 - Distortion Coeffs (k1,k2,p1,p2): 0.0,0.0
- Visualization:**
 - Show Zones

At the bottom, there are buttons for 'Save Config', 'Load Config', and 'Run Detection'.

Розпізнавання автомобілів та визначення їх швидкості



Висновок

У результаті роботи реалізовано метод визначення швидкості транспортних засобів у відеопотоці з використанням YOLOv10n та DeepSORT. Система точно визначає координати, обчислює швидкість з урахуванням параметрів камери та забезпечує візуалізацію результатів.

Експериментальна перевірка на контрольній ділянці автомагістралі (50 м) підтвердила точність методу — виміряні значення швидкості виявили високу кореляцію з еталонними, отриманими вручну за кількістю кадрів (при FPS = 30).

Отримані метрики (precision = 0.86, recall = 0.73, mAP@0.5 > 0.83) демонструють надійність виявлення навіть в умовах ускладненої зйомки.

Розроблений застосунок готовий до інтеграції в системи дорожнього моніторингу та має значний потенціал для масштабування й подальшого розвитку.

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 3.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 14%

ID: 246680 Title: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами Added in a DB: 2025-06-18 Authors: Микола МЕЛЬНИК Heads: Руслан БАГРІЙ Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	69218	1036	4061 (6%)	60 (6%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Микола МЕЛЬНИК

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами

Науковий керівник: Руслан БАГРІЙ, к.т.н., доцент

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1: 5.2%

Коефіцієнт подібності 2: 2.1%

Мікропробіли: 1

Заміна букв: 11

Інтервали: 0

Білі знаки: 3

Дата створення звіту: 2025-06-18 10:39:53.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-18

Дата

експерт

Левко Сергій Р. Р.

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами

Автор студент групи КН-21-1 Микола Мельник

Освітня програма Комп'ютерні науки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 – Комп'ютерні науки

Науковий керівник: к.т.н., доц. каф. комп'ютерних наук Руслан Багрій

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<i>відповідає</i>
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	<i>відсутні</i>

Підтвердження:

Запозичення, виявлені в роботі Миколи Мельника, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.

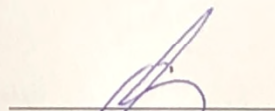
Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:

- за системою Anti-Plagiarism: 3%;

- за системою StrikePlagiarism КП1: 5.21%, КП2: 2,1%.

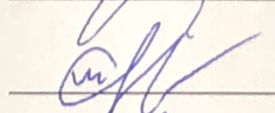
17.06.2025

Завідувач кафедри



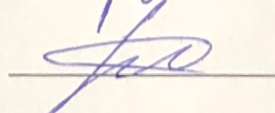
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Руслан БАГРІЙ



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента гр. КН-21-1 Мельника Миколи Романовича

за темою Метод визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами

1. Актуальність теми

Тема роботи є актуальною через зростаючу потребу в інтелектуальних системах для аналізу відеопотоків, зокрема в контексті забезпечення безпеки дорожнього руху. Розробка методу визначення швидкості автомобілів у реальному часі з використанням нейромережових технологій, таких як донавчана модель YOLO, відповідає сучасним вимогам до автоматизації та оптимізації транспортних процесів.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Робота відповідає предметній області спеціальності 122 Комп'ютерні науки, відображаючи практичне застосування сучасних методів обробки даних. У ній реалізовано метод, що включає детекцію об'єктів, їх відстеження в кадрах і обчислення швидкості на основі аналізу відеопотоку. Використано сучасні методи, зокрема нейронні мережі та донавчання, що відповідає вимогам до кваліфікаційних робіт бакалавра.

3. Професійні та особистісні якості бакалавра

Мельник М. Р. проявив високий рівень дисципліни, аналітичних здібностей та практичних навичок під час роботи над проєктом. Його вміння працювати з нейромережевими моделями та інтерпретувати експериментальні дані свідчать про професійну підготовку та ініціативність.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійно з дотриманням принципів академічної чесності. Усі запозичені матеріали та методи правильно оформлені з посиланнями, що підтверджує оригінальність підходу та відсутність плагіату.

5. Ступінь оволодіння методами дослідження

Студент продемонстрував ґрунтовне розуміння підходів до обробки відеоданих та технологій нейромережових систем, орієнтованих на загальну ідентифікацію об'єктів у відео. Його компетентність у роботі з метриками та аналізі результатів

тестування підкреслює високий рівень оволодіння сучасними інструментами комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема розкрита повноцінно: проведено огляд існуючих підходів, розроблено метод визначення швидкості на основі моделі YOLO, виконано експериментальне тестування з аналізом графіків (Precision-Recall, матриці помилок) та таблиць із метриками. Результати підтверджують високу якість методу в реальних умовах.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Матеріал викладено структуровано, із логічною послідовністю від теоретичного аналізу до практичної реалізації. Аргументація базується на експериментальних даних, а стиль відповідає науковим стандартам, забезпечуючи чіткість і професійність викладу.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Метод має практичне значення для систем відеоаналітики, зокрема для контролю швидкості на дорогах і забезпечення безпеки. Окремі елементи, такі як детекція чи відстеження об'єктів, можуть бути адаптовані для інших завдань у сфері комп'ютерного зору.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання, повноту розкриття теми та дотримання всіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка – «відмінно».

Керівник _____ к.т.н., доц. Руслан Багрій



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-21-1 Мельника Миколи Романовича

за темою: Метод визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами

1. Актуальність обраної теми

У сучасних умовах розвитку інтелектуальних систем відеоаналітики велике значення має автоматизація процесів контролю дорожнього руху. Тема роботи є актуальною, оскільки запропонований метод дозволяє визначати швидкість транспортних засобів у реальному часі на основі відеопотоку, що може бути використано для підвищення безпеки та ефективності на дорогах.

2. Повнота розкриття мети та завдань роботи

У кваліфікаційній роботі досягнуто поставлену мету — підвищено точність визначення швидкості руху автомобілів у відеопотоці за допомогою нейромережеских засобів. Усі завдання виконано: розроблено підхід до виявлення транспортних засобів, реалізовано механізм їх відстеження в кадрі та здійснено обчислення швидкості на основі аналізу пройденої відстані та часу.

3. Зміст кожного розділу роботи

Робота складається з трьох розділів. Перший розділ містить огляд сучасних підходів до детекції та відстеження об'єктів на відео, а також аналіз існуючих методів визначення швидкості. Другий розділ містить опис проєктування методу визначення швидкості руху автомобіля у відеопотоці нейромережевими засобами. Третій розділ присвячено експериментальному тестуванню, оцінці точності методу.

4. Оцінка розробленого методу та його практична цінність

Розроблена система має високу практичну значущість, оскільки може застосовуватися для автоматизованого контролю швидкості на дорогах та безпеки дорожнього руху. Результати експериментів свідчать про точність і ефективність методу в умовах реального відео.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота оформлена згідно з вимогами, має чітку структуру, логічну послідовність викладення матеріалу, наявні всі необхідні ілюстрації, графіки та таблиці.

6. Недоліки кваліфікаційної роботи бакалавра

До недоліків можна віднести обмежену кількість відеофрагментів, на яких проводилось тестування. Також можливе подальше удосконалення методу шляхом підвищення стійкості до складних погодних умов.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка Відмінно.

Рецензент

Андрій Рамаскив