МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА
Факультет економіки і підприємництва
Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2025 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

РОЗРОБКА ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ШВИДКОСТІ ТРАНСПОРТНИХ ЗАСОБІВ

Виконав – студент 4 курсу, групи 41-кн
освітньої програми "Комп’ютерні науки"
спеціальності 122 Комп’ютерні науки
Штерц Ростислав Олегович

Науковий керівник – к.т.н., доцент
Ліщук Роман Ігорович

Умань – 2025

Зміст

ВСТУП 3

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ ШВИДКОСТІ РУХУ ТРАНСПОРТНИХ ЗАСОБІВ 6

1.1. Аналіз сучасних систем для розпізнавання швидкості автомобілів. 6

1.2. Огляд наявних аналогів 9

1.3. Постановка задачі 14

РОЗДІЛ 2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ШВИДКОСТІ ТРАНСПОРТНИХ ЗАСОБІВ 17

2.1. Аналіз мов програмування та середовищ розробки 17

2.2. Математичне моделювання процесу розпізнавання швидкості транспортних засобів з використанням YOLO 19

2.3. Архітектура нейронної мережі YOLO для розпізнавання транспортних засобів та визначення їх швидкості 22

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ШВИДКОСТІ ТРАНСПОРТНИХ ЗАСОБІВ 26

3.1. Підготовка вхідних даних 26

3.2. Розпізнавання транспортних засобів і визначення їх швидкості з використанням YOLOv7 28

3.3. Особливості модернізації YOLOv7 для розпізнавання швидкості транспортних засобів 31

ВИСНОВКИ 36

СПИСОК ЛІТЕРАТУРИ 38

ДОДАТКИ ..... **Помилка! Закладку не визначено.**

ВСТУП

Розвиток інтелектуальних транспортних систем (ITC) вимагає створення надійних і точних засобів моніторингу дорожнього руху, серед яких особливої уваги заслуговує вимірювання швидкості транспортних засобів. Порушення швидкісного режиму є однією з основних причин дорожньо-транспортних пригод, що зумовлює необхідність впровадження автоматизованих систем контролю. Традиційні методи (радарні установки, індукційні петлі, лідари) мають низку обмежень: високу вартість, складність монтажу, обмежену масштабованість. У цьому контексті використання комп'ютерного зору та методів глибокого навчання відкриває нові перспективи для розробки адаптивних, ефективних та недорогих систем спостереження, здатних у режимі реального часу визначати швидкість та інші характеристики транспортних засобів на основі відеопотоку[1-5].

Проблематика дослідження активно розробляється в межах машинного навчання, комп'ютерного зору та транспортної інформатики. Значний внесок у розвиток методів розпізнавання та трекінгу об'єктів зробили дослідники Redmon і Farhadi (YOLO), Wojke (DeepSORT), Zhang (ByteTrack). Також актуальними є роботи в галузі відеоаналітики дорожнього руху, зокрема у межах досліджень IEEE Intelligent Transportation Systems та ACM Computing Surveys[6-8].

Метою даної роботи є розробка інформаційної системи для розпізнавання швидкості транспортних засобів на основі методів комп'ютерного зору з використанням нейронної мережі YOLOv7 та алгоритмів трекінгу об'єктів.

Об'єкт дослідження – процес відеоаналізу дорожнього руху в умовах відкритого середовища з використанням оптичного спостереження.

Предмет дослідження – методи та програмні засоби розпізнавання транспортних засобів і визначення їхньої швидкості за допомогою алгоритмів комп'ютерного зору.

Методи дослідження:

- методи математичного аналізу та статистики – для аналізу ефективності обчислень і точності системи;

- методи комп’ютерного зору – для розпізнавання та трекінгу об’єктів;
- нейронні мережі (YOLOv7) – для виявлення транспортних засобів на відео;
- алгоритми трекінгу (наприклад, DeepSORT, ByteTrack) – для побудови траєкторій руху транспортних засобів;
- програмна реалізація засобами Python, OpenCV, PyTorch – для створення експериментального прототипу системи.

Результати цієї роботи можуть бути застосовані в системах автоматичного контролю швидкості, інтелектуального відеоспостереження, аналітики дорожнього трафіку та безпеки руху.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ ШВИДКОСТІ РУХУ ТРАНСПОРТНИХ ЗАСОБІВ

1.1. Аналіз сучасних систем для розпізнавання швидкості автомобілів

У процесі еволюції транспортної інфраструктури визначення швидкості транспортних засобів стало одним із ключових завдань автоматизованих систем дорожнього моніторингу. Точне вимірювання швидкості руху є важливим для забезпечення безпеки дорожнього руху, дотримання правил, запобігання порушенням, збору аналітичної інформації для транспортного планування, а також формування адаптивних систем управління дорожнім рухом.

Системи для визначення швидкості умовно поділяються на такі категорії: контактні (наприклад, індукційні петлі), безконтактні (оптичні, радіохвильові, лазерні), супутникові (GPS/GLONASS), а також відеоаналітичні системи на основі комп’ютерного зору. Кожен тип має свої переваги та обмеження, що визначають доцільність використання в певних умовах[9-10].

Контактні технології включають індукційні сенсори, встановлені під дорожнім полотном. Швидкість визначається на основі часу проходження транспортного засобу між двома послідовними датчиками. Такі системи демонструють високу точність, проте мають обмежений строк служби, потребують періодичного технічного обслуговування та втручання в дорожню інфраструктуру.

Радарні та LIDAR (Light Detection and Ranging) системи широко використовуються в мобільних і стаціонарних комплексах контролю швидкості. Радарні пристрої застосовують ефект Доплера: зміна частоти відбитого сигналу пропорційна швидкості руху. Лідари, в свою чергу, використовують імпульсне лазерне випромінювання і визначають швидкість за часом повернення відбитого сигналу[11-14]. Формально це подається як:

$$v = \frac{c \cdot \Delta f}{2 f_0 \cos(\theta)},$$

де c – швидкість світла, Δf – зсув частоти, f_0 – початкова частота, θ – кут між напрямком руху об'єкта та напрямком променя.

Попри високу точність, ці системи мають обмеження щодо зон контролю (вузький сектор), високі вимоги до розташування, а також високу вартість лазерного обладнання.

Інфрачервоні та оптичні сенсори реалізуються у вигляді бар’єрів або рамок. Перетин променя транспортним засобом активує таймери, і швидкість визначається з відомої довжини ділянки. Обчислення виконується за базовою формулою:

$$v = \frac{l}{t},$$

де l – відстань між сенсорами, t – час проходження.

Такі системи дешеві, проте вразливі до погодних умов, не дозволяють проводити класифікацію транспортних засобів і мають складнощі в ідентифікації при щільному русі.

Системи супутникового моніторингу (GPS/GLONASS) застосовуються у транспортній логістиці, пасажирському та вантажному транспорті. Принцип базується на розрахунку швидкості як похідної зміни просторового положення:

$$v = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{t_2 - t_1}.$$

Незважаючи на зручність, GPS має обмеження в точності при низькій частоті оновлення або втраті сигналу (тунелі, урбанізовані зони), що робить його неефективним для зовнішнього контролю швидкості.

Комп’ютерний зір є новітнім напрямом, що поєднує оптичне спостереження та методи глибокого навчання. Системи цього типу не лише визначають швидкість, але й дозволяють детектувати тип транспортного засобу, розпізнавати номерні знаки, виконувати трекінг об’єктів[15-20]. Процес включає кілька етапів:

1. Детекція об’єктів – зазвичай використовується CNN (зокрема YOLO, SSD, Faster R-CNN), що дозволяють у реальному часі виявити транспортні засоби на відеопотоці.
2. Трекінг – відстеження об'єктів у часі (наприклад, за допомогою DeepSORT або ByteTrack), що дає змогу зафіксувати їх положення в різні моменти часу.
3. Розрахунок швидкості здійснюється на основі відстані між віртуальними контрольними лініями та кількості кадрів між перетинами:

$$v = \frac{d \cdot f}{i_2 - i_1},$$

де d – відстань між віртуальними лініями (в метрах), f – частота відеопотоку (кадрів/сек), i_1, i_2 – кадри перетину.

Ключовою перевагою є можливість масштабування, обробки великої кількості об’єктів одночасно та інтеграції з іншими функціями (наприклад, визначення напрямку руху, порушень ПДР). Разом із тим, ефективність таких систем залежить від калібрування камери, стабільності відеопотоку, умов освітлення та якості алгоритмів трекінгу.

Порівняльна характеристика основних типів систем вимірювання швидкості

Тип системи	Точність	Вартість	Масштабованість	Складність встановлення	Можливість класифікації
Індукційні петлі	Висока	Середня	Низька	Висока	Немає
Радар	Висока	Середня	Середня	Середня	Немає
Лідар	Дуже висока	Висока	Низька	Висока	Обмежена
Інфрачервоні датчики	Середня	Низька	Низька	Низька	Немає
GPS	Середня	Середня	Висока	Низька	Так
Комп’ютерний зір	Висока	Висока	Висока	Середня	Так

Системи вимірювання швидкості транспортних засобів суттєво відрізняються за своїми технічними та експлуатаційними характеристиками. Традиційні підходи, хоч і зберігають актуальність, поступово витісняються інтелектуальними відеоаналітичними методами, які забезпечують вищий рівень функціональності та гнучкості. Особливо перспективним є використання систем на базі комп'ютерного зору в поєднанні з алгоритмами глибокого навчання, що дозволяє ефективно вирішувати комплексні задачі: від розпізнавання до класифікації та розрахунку швидкості в реальному часі. Надалі основною тенденцією розвитку є створення гібридних систем, які інтегрують комп'ютерний зір із сенсорними та супутниковими технологіями для підвищення точності та надійності в будь-яких умовах.

1.2. Огляд наявних аналогів

На сучасному етапі існує декілька основних класів систем для вимірювання швидкості транспортних засобів: радарні системи, лазерні далекоміри (LIDAR-системи), індукційні датчики, GPS-моніторинг та системи на базі комп'ютерного зору. Розглянемо детально кожен із підходів із урахуванням п'яти ключових критеріїв: точність, вартість, масштабованість, складність встановлення, можливість класифікації транспортних засобів.

На практиці найбільш розповсюдженими на автошляхах України й країн ЄС є радарні й індукційні системи. Вони використовуються поліцією для фіксації перевищення швидкості, а також у системах автоматизованого збору статистики трафіку. Наприклад, система TruCAM, яка активно використовується в Україні, є портативним лазерним радаром з функцією фіксації зображення порушника.

Система TruCAM – це сучасний лазерний пристрій, що використовується поліцією для контролю швидкості руху транспортних засобів і фіксації порушень правил дорожнього руху. Його основна особливість полягає в тому, що він не просто вимірює швидкість автомобіля, а й одночасно робить фото- і

відеозапис моменту порушення, забезпечуючи документальний доказ у випадку судового розгляду (рис. 1.1).



Рисунок 1.1 – Пристрій TruCAM

Принцип роботи TruCAM базується на використанні технології LIDAR – це метод дистанційного вимірювання, в якому застосовується лазерний промінь. Коли інспектор наводить пристрій на автомобіль, TruCAM випромінює серію лазерних імпульсів, які відбиваються від машини і повертаються до приймача. Вимірюючи час, за який імпульс долає шлях до об'єкта і назад, пристрій обчислює точну відстань. Якщо таких вимірювань зробити кілька за короткий проміжок часу, можна обчислити, з якою швидкістю змінюється положення об'єкта – тобто його швидкість.

На відміну від традиційних радарів, що використовують радіохвилі, лазерний промінь TruCAM набагато вужчий і більш спрямований. Це дозволяє точно визначити, яке саме авто було виміряне, навіть якщо дорога багатосмугова й транспорту багато. У момент фіксації порушення TruCAM автоматично робить високоякісне фото, на якому чітко видно номер автомобіля, його положення на дорозі, а також у кадрі відображається

швидкість, з якою він рухається, час порушення, відстань до об'єкта і навіть координати (якщо пристрій має GPS-модуль)[35].

TruCAM також може знімати відео – від кількох секунд до хвилини, залежно від налаштувань. Це відео зберігається разом із фото на захищеному цифровому носії (найчастіше SD-карті), і може бути використане як доказ під час розгляду справи в суді або при оскарженні штрафу.

Ще одна важлива особливість – наявність GPS-приймача. Завдяки йому кожен зафіксований епізод має точну геолокацію, тобто точно відомо, де саме було здійснено вимірювання. Це виключає можливість маніпуляцій чи помилок з боку інспектора, який використовує пристрій.

TruCAM зазвичай встановлюється на штатив, але може використовуватись і в ручному режимі. У багатьох країнах, зокрема в Україні, такі прилади використовуються на визначених ділянках доріг, де встановлені попереджувальні знаки про фото- та відеофіксацію.

Основні недоліки – необхідність участі оператора, обмеження у фокусній відстані та відсутність можливості обробки великого потоку ТЗ одночасно.

У деяких містах Європи (наприклад, у Німеччині, Нідерландах) встановлено стаціонарні автоматизовані радарні комплекси з функцією фіксації перевищення швидкості, фотографування та автоматичної передачі даних до поліції. Такі системи дорогі в реалізації й потребують ретельного калібрування, але забезпечують високу точність вимірювання.

Індукційні системи часто використовуються в міських умовах, особливо на світлофорах і перехрестях, для підрахунку транспортного потоку. Їх основною перевагою є низька ймовірність хибних спрацювань у складних погодних умовах. Проте відсутність можливості класифікації ТЗ і складність обслуговування внаслідок розміщення в дорожньому полотні обмежує їхню ефективність.

У сфері вантажних і службових перевезень значного поширення набули GPS-системи контролю швидкості, зокрема продукти компаній Teltonika, Wialon, Omnicomm. Вони дозволяють у реальному часі контролювати

швидкість, маршрут, час простою та місце знаходження транспорту. Проте такі системи обмежені в застосуванні до приватних осіб і вимагають встановлення спеціального обладнання на кожному ТЗ.

Система Teltonika – це комплексне телематичне рішення, створене для того, щоб забезпечити повний контроль і моніторинг транспортних засобів, техніки або будь-яких рухомих об’єктів у режимі реального часу. Вона поєднує у собі спеціалізовані GPS-пристрої (трекери), розроблені компанією Teltonika, програмне забезпечення для обробки та візуалізації даних, а також різноманітні допоміжні сенсори та модулі, що дозволяють адаптувати систему під конкретні потреби бізнесу або індивідуального користувача.

Принцип дії цієї системи досить простий, але технічно досконалий. У транспортний засіб (чи будь-який інший об’єкт) встановлюється спеціальний трекер (рис. 1.2.) – це невеликий пристрій, що підключається до живлення авто і, за необхідності, до CAN-шини чи інших систем транспортного засобу. Основне завдання трекера – визначати точні координати об’єкта за допомогою супутникових систем позиціонування, таких як GPS, GLONASS, Galileo або BeiDou. Але на цьому його функції не обмежуються.



Рисунок 1.2 – GPS-трекер Teltonika FMB125

Ці трекери також збирають велику кількість додаткових даних: швидкість руху, час у дорозі, моменти зупинки, стан запалювання, поведінку водія

(наприклад, різкі гальмування, прискорення, повороти). Якщо пристрій підключено до CAN-шини автомобіля, він може зчитувати ще більше технічних параметрів – витрати пального, оберти двигуна, температуру, пробіг, відкриття дверей, помилки ЕБУ тощо.

Вся ця інформація передається на віддалений сервер через мобільний зв'язок (GSM/3G/4G), де вона обробляється, зберігається й візуалізується за допомогою спеціального програмного забезпечення. Для цього Teltonika пропонує власні платформи або підтримує інтеграцію з популярними системами, такими як Wialon, GPSWOX, Naviky та іншими. У цих програмах користувач може в реальному часі бачити місцезнаходження кожного транспортного засобу на карті, переглядати історію маршрутів, отримувати звіти, аналізувати ефективність роботи водіїв, планувати технічне обслуговування і виявляти порушення або підозрілі ситуації.

Однією з важливих переваг системи є можливість працювати з додатковими модулями. Наприклад, якщо потрібно контролювати витрати пального, можна підключити спеціальні датчики, які фіксують обсяг пального в баку з високою точністю. Для контролю доступу до автомобіля можна встановити RFID або iButton зчитувач – тоді кожен водій буде ідентифікований, і можна буде побачити, хто і коли керував конкретною машиною. У сфері вантажоперевезень використовуються температурні сенсори для холодильних установок, щоб гарантувати дотримання температурного режиму при транспортуванні продуктів або медикаментів.

Ще однією важливою особливістю є наявність Bluetooth у багатьох моделях трекерів Teltonika. Це дозволяє підключати бездротові датчики або навіть смартфони водіїв – для додаткової аутентифікації або зручного збору даних. Наприклад, можна реалізувати функцію електронного обліку часу роботи водія, відстежувати входження в геозони, або навіть блокувати запуск двигуна, якщо система не ідентифікує водія[36].

Система активно використовується в логістиці, агробізнесі, будівельній галузі, службах доставки, муніципальному транспорті, таксі та навіть у

військовій сфері. Одна з ключових цілей – це оптимізація витрат та підвищення прозорості в управлінні автопарком. Наприклад, компанія може зменшити витрати на пальне, уникнути шахрайства з боку водіїв, підвищити дисципліну, краще планувати маршрути та уникати зайвого зносу техніки.

Завдяки гнучкості налаштувань, обладнання Teltonika підходить як для малого бізнесу, так і для масштабних підприємств з десятками або сотнями одиниць техніки. Система може бути повністю автоматизована: у випадку фіксації порушення, витоку пального або позапланового виїзду – відповідальна особа одразу отримує сповіщення через e-mail або у вигляді push-сповіщення на телефон.

Таким чином, Teltonika – це не просто GPS-трекери. Це повноцінна платформа для управління мобільними ресурсами, яка дає змогу бачити все, що відбувається з транспортом у реальному часі, виявляти ризики, приймати зважені рішення та підвищувати ефективність у будь-якому бізнес-процесі, де залучений транспорт.

1.3. Постановка задачі

На основі проведеного аналізу можна зробити висновок про доцільність створення автоматизованої системи для визначення швидкості транспортних засобів, що ґрунтується на методах комп’ютерного зору та глибокого навчання. Зокрема, вибір архітектури YOLOv7 (You Only Look Once, версія 7) як основного детектора об’єктів обумовлений її високою швидкістю обробки кадрів у реальному часі, точністю локалізації об'єктів та ефективністю на відеопотоках з обмеженими ресурсами.

Суть задачі полягає у виявленні транспортних засобів на відеозаписі, відстеженні їх руху між заздалегідь визначеними лініями (наприклад, A і B, що перпендикулярні до дороги) та розрахунку швидкості за наступною формулою:

$$v=S*d/t$$

де:

- v – швидкість транспортного засобу,

- S – відстань між положеннями об’єкта у пікселях,
- t – час між кадрами,
- d – масштабний коефіцієнт (кількість метрів на один піксель, визначається калібруванням сцени).

Для цього система повинна реалізовувати наступні пункти:

1. Детекція транспортних засобів. Виявлення об'єктів транспортної категорії (легкові, вантажні автомобілі, автобуси, мотоцикли тощо) на кожному кадрі відеопотоку за допомогою моделі YOLOv7.
2. Трекінг об’єктів. Побудова та підтримка траєкторій руху виявлених об’єктів з використанням одного з сучасних трекерів (наприклад, DeepSORT, ByteTrack або OC-SORT).
3. Маркування контрольних ліній. Розміщення двох або більше логічних ліній на кадрі, перетин яких дозволяє здійснити фіксацію моментів входу і виходу транспортного засобу з контрольованої зони.
4. Обчислення швидкості. На основі часової мітки перетину ліній системою проводиться розрахунок часу $\upsilon = d/t$, після чого застосовується формула з врахуванням масштабу відео (кількості метрів на піксель).
5. Формування звіту. Створення логів з результатами: тип транспортного засобу, швидкість, чи було перевищення швидкості.

Формалізовано задачу можна описати так:

1. Відеопотік з фіксованої камери, що спостерігає за дорогою під кутом, який дозволяє бачити рух транспортних засобів уздовж контрольованої траєкторії.

2. Набір даних для кожного транспортного засобу.

3. Побудова адаптивної, масштабованої системи, здатної працювати в реальному часі на стандартних відеопотоках із камер спостереження без потреби в додаткових сенсорах.

Обмеження**:**

- Камера має бути зафіксована жорстко без зміни положення;

- Зона спостереження має бути заздалегідь калібрована (задано реальну довжину між лініями);
- Необхідно враховувати перспективну трансформацію для коректного співвідношення пікселів до метрів.

У результаті реалізації поставленої задачі очікується створення інформаційної системи з такими характеристиками:

- Автоматичне визначення швидкості;
- Висока точність розрахунків (похибка не більше 10%);
- Можливість розширення функціоналу (ідентифікація номерного знака, побудова статистики тощо);
- Відкритість для масштабування на різні типи доріг і умови.

Таким чином, розробка інформаційної системи на базі YOLO дозволяє ефективно поєднати сучасні методи комп’ютерного зору з практичними вимогами дорожнього моніторингу в умовах міської й міжміської інфраструктури.

РОЗДІЛ 2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ШВИДКОСТІ ТРАНСПОРТНИХ ЗАСОБІВ

2.1. Аналіз мов програмування та середовищ розробки

При створенні системи для розпізнавання швидкості транспортних засобів критично важливим є грамотний вибір мови програмування та відповідного середовища розробки. Вибір цих інструментів обумовлюється вимогами до високої продуктивності, здатності працювати з потоковим відео, реалізації алгоритмів комп'ютерного зору, нейронних мереж і обробки великих обсягів даних у реальному часі. Крім того, важливими є такі характеристики, як підтримка актуальних бібліотек, спільнота користувачів, адаптованість до масштабування і супутня інфраструктура.

На основі вищезгаданих критеріїв для реалізації поставленої задачі було обрано мову програмування Python. Це мова загального призначення, яка завдяки простому синтаксису та великій кількості бібліотек стала стандартом де-факто в галузях машинного навчання, аналізу даних, комп'ютерного зору та штучного інтелекту.

Python вирізняється високим рівнем абстракції, що дає змогу зосередитися безпосередньо на реалізації функціональної логіки, не витрачаючи зайвий час на низькорівневі операції. Більшість сучасних алгоритмів і фреймворків глибокого навчання реалізовані саме під Python, зокрема такі як PyTorch, TensorFlow, Keras, Detectron2, MMDetection та інші. В контексті даного дослідження використовується PyTorch як базовий фреймворк, а також модель YOLOv7, яка має офіційну реалізацію на Python через бібліотеку Ultralytics[21-23].

Важливою перевагою Python є наявність бібліотек, орієнтованих саме на обробку зображень та відео. Зокрема, OpenCV (Open Source Computer Vision Library) дозволяє виконувати такі операції, як зчитування відео з камери, детекція об'єктів, фільтрація зображень, побудова масок, ліній, контурування, а

також конвертація кольорових просторів. Ці інструменти є базовими для реалізації фреймворку виявлення швидкості.

Для реалізації системи трекінгу – необхідного етапу, який дозволяє простежити рух транспортного засобу між двома точками – також застосовуються Python-бібліотеки. Найбільш поширеними є Deep SORT, ByteTrack, StrongSORT та OC-SORT. Всі вони мають відкриті реалізації на Python та здатні інтегруватися безпосередньо з виходами моделей YOLO.

Python має потужну інфраструктуру для наукових обчислень і статистичної обробки даних. Бібліотеки NumPy та Pandas забезпечують швидку роботу з масивами, таблицями, збереженням і аналізом великих обсягів інформації, що генерується під час обробки відео. Matplotlib і Seaborn дозволяють візуалізувати отримані результати, побудувати графіки, гістограми розподілу швидкостей, теплові карти порушень тощо[24-26].

Ще однією перевагою Python є його широка сумісність із зовнішніми інтерфейсами, REST API, хмарними сервісами та базами даних. Це дає змогу легко створити веб-інтерфейс системи, зберігати інформацію у базах (MySQL, MongoDB, SQLite), інтегруватися з серверними службами чи розгортати систему у хмарному середовищі (наприклад, на сервері AWS або Google Cloud Platform).

У якості середовища розробки пропонується використання Jupyter Notebook або Visual Studio Code (VS Code). Перший підходить для швидкої розробки, тестування, демонстрацій та візуалізації проміжних етапів, а другий – для структурованої, модульної розробки з підтримкою розширень, дебагінгу та підключення до Git.

Альтернативні мови програмування також були проаналізовані. Наприклад, C++ дає змогу досягнути високої продуктивності, однак ускладнює процес реалізації, потребує більше часу на розробку та не має такого широкого спектру готових бібліотек для глибокого навчання. Крім того, інтеграція C++ із сучасними моделями глибокого навчання часто вимагає додаткових обгорток або компіляції зовнішніх залежностей.

Іншим варіантом може бути використання JavaScript у зв'язці з TensorFlow.js, проте обробка відео в реальному часі у браузері має обмеження щодо продуктивності. Подібна система могла б функціонувати лише на стороні клієнта у веб-додатку, але це не задовольняє вимоги до реального часу, точності та стабільності [27-30].

Також було розглянуто використання MATLAB, який часто застосовується для наукових досліджень і візуалізації. Проте закритість платформи, висока вартість ліцензій та слабка підтримка для інтеграції в реальні системи обмежують доцільність його використання у даному проєкті.

Окрім вибору мови, критично важливо забезпечити підтримку контейнеризації для легкого розгортання системи. Python має підтримку Docker, що дозволяє створити незалежне середовище з усіма необхідними залежностями, запускати систему на різних серверах, розширювати кількість відеопотоків та масштабувати архітектуру. Також, Python сумісний з Kubernetes та іншими засобами оркестрації контейнерів, що робить його придатним для розгортання в умовах хмарних обчислень або локальних серверів [30-32].

Таким чином, мова програмування Python є найкращим вибором для реалізації інформаційної системи розпізнавання швидкості транспортних засобів на базі YOLOv7. Вона забезпечує простоту, гнучкість, високу сумісність з існуючими фреймворками машинного навчання, розширюваність і відповідність усім технічним вимогам до побудови систем реального часу з обробкою відео, детекцією, трекінгом і розрахунком швидкості.

2.2. Математичне моделювання процесу розпізнавання швидкості транспортних засобів з використанням YOLO

Розпізнавання швидкості транспортних засобів на відеопотоці з камери спостереження є складною задачею комп'ютерного зору, що поєднує методи розпізнавання об'єктів, трекінгу та обчислення кінематичних параметрів. У межах цього дослідження для задачі розпізнавання використовується архітектура YOLOv7, яка демонструє високу точність виявлення об'єктів у

реальному часі. Математичне моделювання процесу побудоване на основі аналізу координат об’єктів, які розпізнаються на послідовних кадрах відео.

YOLO – це нейронна мережа для розпізнавання об’єктів, що передбачає координати об’єкта, клас, а також ймовірність присутності об’єкта в межах кожної комірки сітки.

Нехай зображення має розмір W×H пікселів і розбите на сітку розміром S×S. Кожна клітинка прогнозує B обмежувальних прямокутників (bounding boxes), кожен з яких характеризується:

$$y = [x, y, w, h, P_{obj}, p_1, p_2, ..., p_C]$$

де:

x,y – координати центра боксу (відносно клітинки),

w, h – ширина та висота (відносно всього зображення),

P_{obj} – ймовірність того, що в боксі є об’єкт,

p_i – ймовірність належності до класу i, i=1..C.

Після успішної розпізнавання транспортного засобу необхідно відстежити його переміщення у часі. Враховуючи дві контрольні лінії на відео, що перетинають дорогу перпендикулярно напрямку руху, швидкість автомобіля визначається за формулою:

$$v = \frac{d}{\Delta t}$$

де:

Δd – фізична відстань між двома контрольними лініями (визначається під час калібрування, в метрах),

$\Delta t = t_2 - t_1$ – час, за який автомобіль пройшов від першої лінії до другої.

Для кожного кадру відео, позначеного як I_t, мережа YOLOv7 повертає множину box-координат:

$$\mathcal{B}_t = \{b_1^t, b_2^t, ..., b_n^t\}$$

Кожен бокс має координати $[x_t,y_t,w_t,h_t]$, що описують положення розпізнаного автомобіля. Центр обмежувальної рамки:

$$C_t = (x_t + \frac{w_t}{2}, y_t + \frac{h_t}{2})$$

Контрольна лінія описується як відрізок у координатах пікселів:

$$L_1 = \{(x_a, y_a), (x_b, y_b)\}$$

Для визначення моменту перетину лінії L_1, використовується аналіз положення центру C_t на поточному кадрі й попередньому:

якщо:

(C_{t-1} знаходився до лінії)∧(C_t знаходиться після лінії)

то вважається, що об'єкт перетнув лінію на кадрі t.

Запам’ятовується значення часу t_1 – перетин першої лінії, та t_2 – перетин другої. Якщо відео має частоту f кадрів на секунду, то:

$$\Delta t = \frac{t_2 - t_1}{f}$$

Після чого швидкість обчислюється:

$$v = \frac{d}{\Delta t} = \frac{d \cdot f}{t_2 - t_1}$$

Оскільки піксельна відстань між лініями на зображенні залежить від кута огляду, фокусної відстані камери та перспективної деформації, потрібна попередня калібровка. Для цього визначається коефіцієнт масштабу k:

$$k = \frac{d}{d_{pix}}$$

де:

d – реальна відстань між лініями (у метрах);

d_{pix} – відстань між лініями в пікселях.

Таким чином, шлях, пройдений об’єктом у пікселях, масштабується:

$$v = \frac{k \cdot d_{pix}}{\Delta t}$$

У разі фіксованого положення камери, коефіцієнт k залишається сталим.

Під час тренування YOLOv7 оптимізує функцію втрат, що складається з трьох компонентів:

$$L = \lambda_{box} \cdot L_{box} + \lambda_{obj} \cdot L_{obj} + \lambda_{cls} \cdot L_{cls}$$

де:

L_{box} – втрата локалізації ,

L_{obj} – втрата на наявність об’єкта (об’єктність),

L_{cls} – втрата класифікації,

λ_i – вагові коефіцієнти [34].

Загальна схема розрахунку швидкості

1. Вхідний відеопотік обробляється з частотою f кадрів/сек.
2. На кожному кадрі виконується детекція транспортних засобів за допомогою YOLOv7.
3. Центри об'єктів відстежуються через кадри.
4. Фіксуються моменти перетину першої та другої лінії.
5. Обчислюється швидкість за формулою:

Математичне моделювання системи розпізнавання швидкості базується на поєднанні геометричної обробки координат bounding box, часових міток кадрів, і просторового калібрування. Завдяки високій точності розпізнавання та здатності до роботи в реальному часі, архітектура YOLOv7 забезпечує ефективну реалізацію системи моніторингу дорожнього руху.

2.3. Архітектура нейронної мережі YOLO для розпізнавання транспортних засобів та визначення їх швидкості

YOLO (You Only Look Once) – це одна з ефективних архітектур (рис. 1.3.) нейронних мереж для задач розпізнавання об’єктів у реальному часі. Зокрема, YOLOv7 є найновішим варіантом із серії YOLO, який пропонує оптимальний баланс між точністю, швидкістю та продуктивністю на GPU. Архітектура YOLOv7 має модульну структуру і складається з трьох основних компонентів: Backbone, Neck і Head.

1. Backbone – модуль для витягування ознак із вхідного зображення. У YOLOv7 зазвичай використовується архітектура на базі CSPDarknet або його модифікації.

CSPDarknet використовує принципи Cross-Stage Partial Network для покращення злиття ознак:

$$X_{l+1} = F([X_l^1, G(X_l^2)])$$

де X_l– вхідний тензор, X_l^1, X_l^2 – його частини, G– набір згорток, F– функція об’єднання.

2. Neck – проміжна частина, що об’єднує ознаки з різних рівнів. У YOLOv7 Neck реалізований через PANet (Path Aggregation Network), що забезпечує передачу як високорівневих, так і низькорівневих ознак:

$$F_{agg} = Concat(Up(f_5), f_4, Down(f_3))$$

Де f_i – ознаки з різних рівнів, Up/Down – операції зміни масштабу, Concat – конкатенація.

3. Head – модуль, що виконує остаточну детекцію. Він формує три карти прогнозів для різних масштабів . Кожна комірка передбачає b-боксів, кожен з яких має координати , ймовірність об’єкта та класи:

$$P(Object) \cdot IoU \cdot P(Class_i)$$

Вихідний тензор має розмірність:

$$S \times S \times (B \cdot (5 + C))$$

де:

- S – розмір сітки;
- B – кількість боксів на клітинку;
- C – кількість класів (у нашому випадку – типи транспортних засобів).

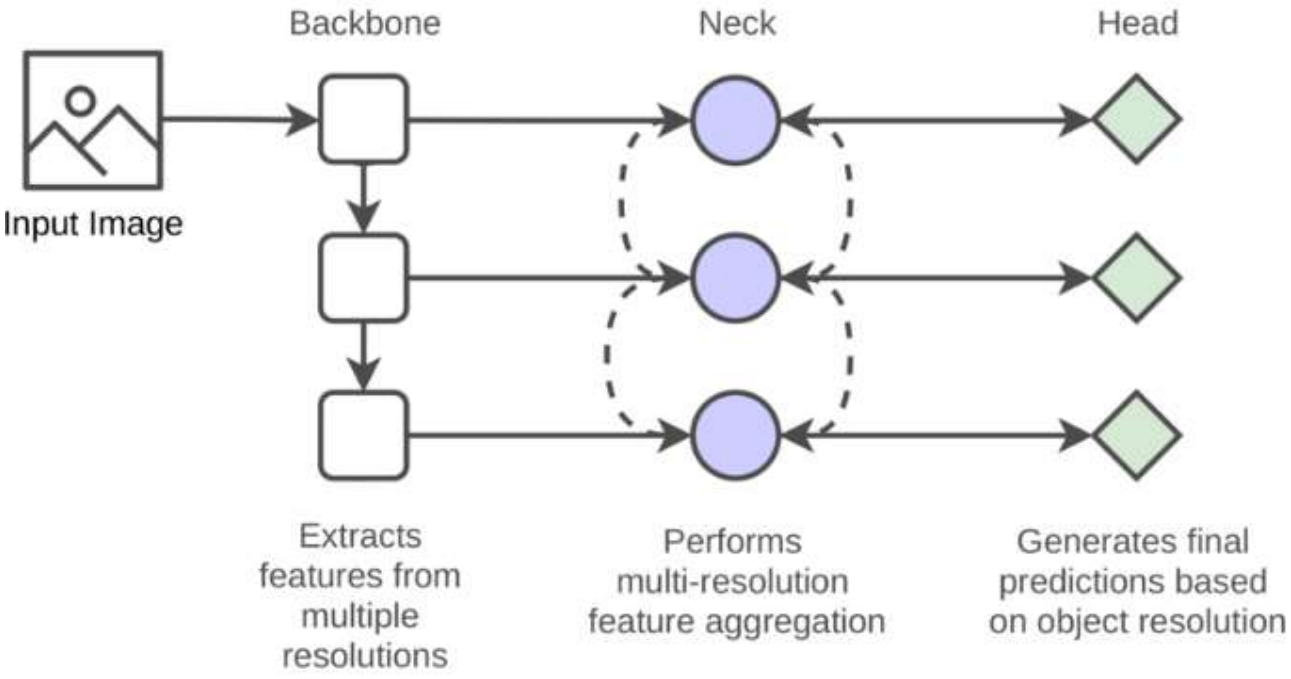


Рисунок 1.3. – Архітектурна схема YOLOv7

Особливості YOLOv7:

- Використання ELAN-модуля (Efficient Layer Aggregation Networks) для покращення передачі ознак;
- RepConv – повторно згортковий блок для поліпшення інференсу;
- Підтримка Auxiliary heads – допоміжні гілки для глибшого навчання;
- Оптимізація за допомогою Label Assignment Techniques .

Кожен бокс передбачає координати центру , ширину , висоту , об'єктність , і ймовірність кожного класу P_{classi} :

$$y = [x, y, w, h, P_{obj}, P_{class_1}, ..., P_{class_C}]$$

Втрата (Loss):

$$L = L_{bbox} + L_{obj} + L_{cls}$$

де L_{bbox} – loss координат, L_{obj} – об'єктність (binary cross-entropy), L_{cls} – класифікаційна помилка [37-38].

У межах цієї роботи YOLOv7 використовується п'ять класів для розпізнавання. Вихідні координати боксів використовуються як базові точки для обчислення швидкості: після проходження транспортним засобом двох

контрольних ліній, зафіксованих у координатах кадру, система вимірює різницю часу і розраховує швидкість:

$$v=S*d/t$$

де:

- v – швидкість транспортного засобу,
- S – відстань між положеннями об’єкта у пікселях,
- t – час між кадрами,
- d – масштабний коефіцієнт.

Завдяки високій точності YOLOv7 і її можливості обробляти відеопотік у реальному часі, ця архітектура є оптимальним вибором для систем розпізнавання швидкості автотранспорту на базі відеоспостереження.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ШВИДКОСТІ ТРАНСПОРТНИХ ЗАСОБІВ

3.1. Підготовка вхідних даних

У процесі побудови системи розпізнавання транспортних засобів на основі моделі YOLOv7 особливе значення має формування якісного та структурованого набору вхідних даних. Надійність і точність роботи моделі великою мірою залежать від репрезентативності датасету, що включає достатню кількість зображень різних типів транспортних засобів у реальних умовах. У цьому підрозділі описано процес створення такого датасету з п’ятьма основними класами: Car, Truck, Bus, Motorcycle, Bicycle.

Перший етап – збір зображень – реалізовувався шляхом об’єднання кількох джерел. Основну частину зображень було отримано з відеозаписів дорожнього руху, знятих як у міських умовах, так і на автомагістралях. Відео оброблялися з використанням спеціальних скриптів, що дозволяли витягувати окремі кадри з частотою 2–3 кадри на секунду. Окрім цього, до набору були додані зображення з відкритих джерел, таких як Open Images Dataset, Berkley DeepDrive та Kaggle.

При формуванні набору особлива увага приділялася різноманітності зображень: було обрано кадри в різні пори року, за різних погодних умов (сонце, дощ, туман), а також за різних кутів огляду. Таке розмаїття сприяє підвищенню стійкості моделі до реальних викликів, зокрема до часткового перекриття транспортних засобів, різної відстані до камери та змін освітлення.

Після збору даних відбувався процес анотування зображень, тобто визначення положення об’єктів і їхніх класів. Для цього використовувалася програма LabelImg, яка дозволяє зручно створювати анотації у форматі YOLO. На кожному зображенні вручну позначалися рамки (bounding boxes) навколо транспортних засобів із вказівкою класу. Кожен клас мав свій числовий ідентифікатор (табл. 3.1).

Таблиця 3.1 Позначення класів

Клас	Позначення	Ідентифікатор
Car	car	0
Truck	truck	1
Bus	bus	2
Motorcycle	motorcycle	3
Bicycle	bicycle	4

Наприклад, запис анотації для об'єкта класу "Truck", що займає центральну частину зображення, виглядатиме так:

1 0.500 0.500 0.300 0.400

де 1 – клас Truck, координати та розміри нормалізовані відносно ширини та висоти зображення.

Всі зображення та відповідні до них анотації були розділені на навчальний і валідаційний набори у співвідношенні 80:20. Для цього структура папок (рис. 3.1) виглядала так:

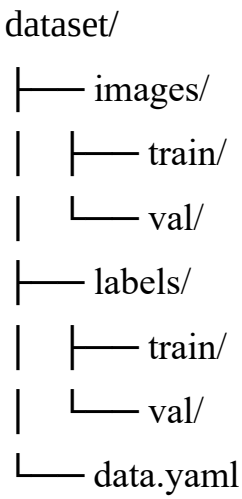


Рисунок 3.1 – Структура папок набору даних

У файлі *data.yaml*, що задає конфігурацію для моделі YOLOv7, було вказано шляхи до зображень та назви класів:

```
train: ./dataset/images/train
val: ./dataset/images/val

nc: 5
names: ['Car', 'Truck', 'Bus', 'Motorcycle', 'Bicycle']
```

Рисунок 3.2 – Приклад файлу *data.yaml*

3.2. Розпізнавання транспортних засобів і визначення їх швидкості з використанням YOLOv7

У межах розробки інформаційної підсистеми для розпізнавання швидкості транспортних засобів важливим етапом є виявлення самих об'єктів контролю, тобто автомобілів, автобусів, вантажівок тощо. Однією з найбільш ефективних технологій для вирішення цього завдання є використання сучасних нейронних мереж для обробки зображень у реальному часі. У цій роботі для реалізації відповідного функціоналу обрано YOLOv7, яка поєднує високу точність розпізнавання з дуже хорошою швидкодією.

YOLOv7 у порівнянні з попередніми версіями (YOLOv5, YOLOv6) має низку переваг. Зокрема, вона включає покращені архітектурні блоки, такі як E-ELAN, та підтримує глибоку оптимізацію для GPU. Завдяки цьому YOLOv7 демонструє найкраще співвідношення між точністю (mAP) та швидкістю (FPS) на момент впровадження.

На рис. 3.3 зображено результат роботи моделі YOLOv7 на одному з кадрів відео міського трафіку.

Як видно з рисунка, модель успішно виявляє кілька типів транспортних засобів та присвоює їм відповідні мітки. Рамки (bounding boxes), отримані від нейронної мережі, використовуються не лише для візуалізації, але й для подальшого аналізу руху.

Після отримання координат виявлених транспортних засобів на кожному кадрі постає завдання їх трекінгу – тобто відстеження одного й того самого об'єкта між послідовними кадрами. Для цього використовується алгоритм Deep

SORT, який враховує як просторові координати об'єкта, так і векторні ознаки (features), отримані з глибокої нейромережі.

Завдяки використанню трекера кожному транспортному засобу присвоюється унікальний ідентифікатор, що дозволяє точно визначити його переміщення на певному часовому проміжку.

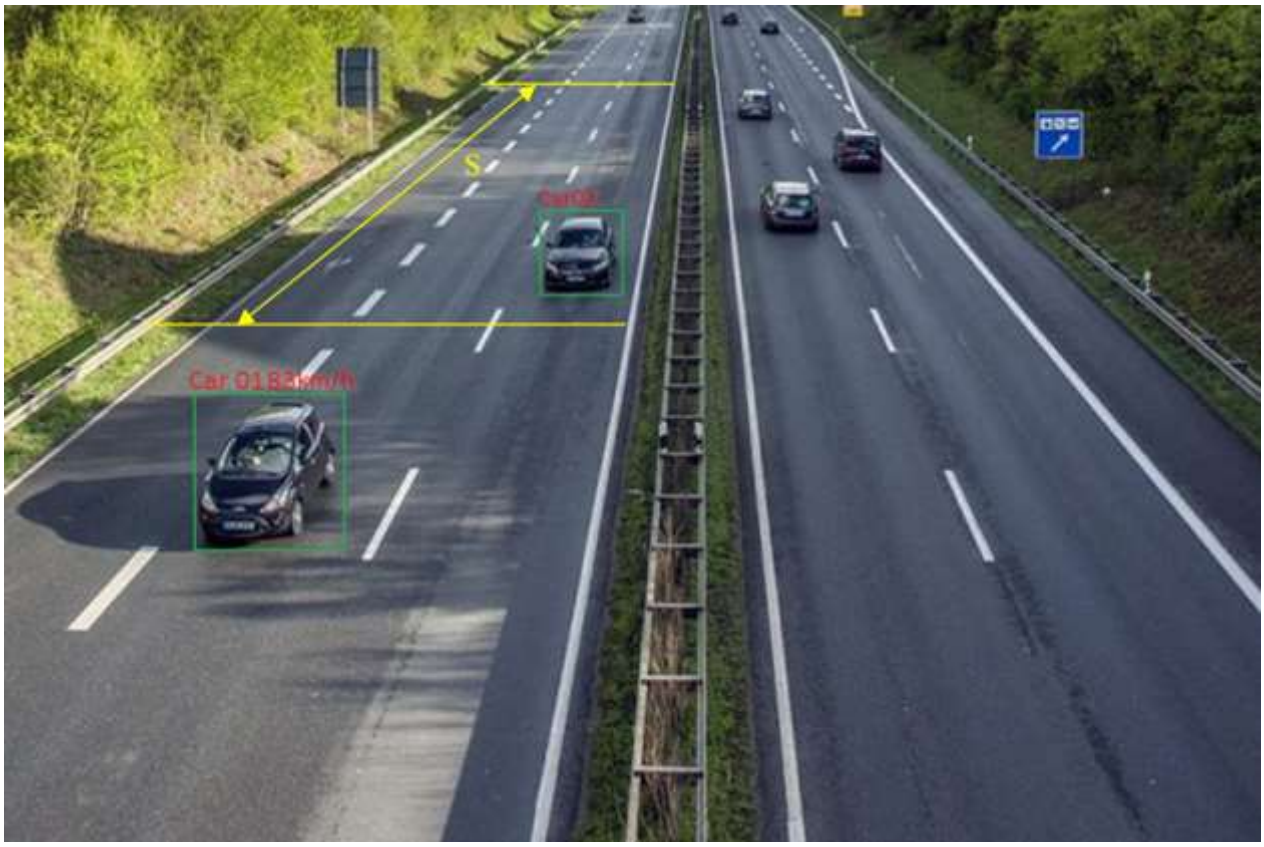


Рисунок 3.3– Результат розпізнавання транспортних засобів та визначення їх швидкості за допомогою YOLOv7

Знаючи координати об’єкта в просторі кадру у різні моменти часу, можливо обчислити його швидкість. Це здійснюється шляхом вимірювання відстані між положеннями транспортного засобу на різних кадрах та поділу цієї відстані на відповідний інтервал часу. Розрахунок здійснюється за формулою:

$$v=S*d/t$$

де:

- v – швидкість транспортного засобу,
- S – відстань між положеннями об’єкта у пікселях,

- t – час між кадрами,
- d – масштабний коефіцієнт (кількість метрів на один піксель, визначається калібруванням сцени).

На рис. 3.3 наведено приклад калібрування дорожньої сцени з використанням маркерів на реальній дорозі для точного визначення масштабу.

Розпізнавання автомобілів за допомогою моделі YOLO починається з розбиття вхідного зображення на сітку з однаковими клітинками, наприклад, розміром 13×13 або 19×19. Кожна така клітинка відповідає за виявлення об’єктів, що можуть перебувати в її межах. Для кожної області сітки нейронна мережа формує кілька варіантів прямокутників (bounding boxes), оцінюючи ймовірність того, що всередині кожного з них знаходиться об’єкт, зокрема автомобіль. Окрім координат прямокутника, кожен із них також містить передбачення належності до певного класу об’єктів. Після первинного аналізу YOLO відбирає ті рамки, у яких імовірність виявлення автомобіля є найвищою. Якщо кілька рамок накладаються на один і той самий об'єкт, застосовується метод пригнічення непотрібних варіантів (Non-Maximum Suppression), який залишає лише найбільш точне та достовірне передбачення. У вдосконаленій версії YOLO до параметрів, які визначають положення рамки та клас об'єкта, було додано ще один – швидкість автомобіля. Це дозволяє не лише виявляти транспортні засоби, але й одразу фіксувати їхню швидкість у момент виявлення. Такий підхід дозволяє ефективно працювати навіть у складних дорожніх умовах: при великій кількості автомобілів, перехрещенні об’єктів, зміні кута огляду або коливаннях освітлення[11].

Отримані швидкості зберігаються у базі даних і порівнюються із допустимими значеннями для відповідної ділянки дороги. У разі виявлення перевищення швидкісного ліміту, система може фіксувати порушення та автоматично генерувати звіт або відповідне повідомлення.

3.3. Особливості модернізації YOLOv7 для розпізнавання швидкості транспортних засобів

YOLOv7 є однією з найефективніших сучасних моделей для виявлення об’єктів у реальному часі, зокрема для задач розпізнавання транспортних засобів на зображеннях і відео. Проте для розширення її функціональності в контексті інтелектуального аналізу дорожньої обстановки постала потреба у модернізації архітектури моделі та її інтеграції з додатковими механізмами, що дозволяють визначати швидкість руху транспортних засобів.

Стандартна архітектура YOLOv7 призначена лише для розпізнавання об’єктів – вона визначає координати прямокутника навколо об'єкта, його клас та ймовірність передбачення. Для оцінки швидкості необхідно відстежити зміну положення транспортного засобу між кількома послідовними кадрами. Отже, постає задача відстеження об'єктів, а також математичного обчислення швидкості на основі переміщення рамки об'єкта у часі.

Щоб правильно обчислити швидкість, необхідно знати точний масштаб сцени, тобто співвідношення між розміром зображення та фізичними відстанями. Це можна зробити за допомогою калібрування камери або ж за допомогою відомих об'єктів у кадрі (наприклад, ширини смуги чи довжини дорожньої розмітки).

Для реалізації визначення швидкості було використано наступний підхід:

1. Детекція об’єктів на кожному кадрі за допомогою моделі YOLOv7.
2. Відстеження одного й того самого об’єкта між кількома кадрами з використанням алгоритму трекінгу (наприклад, DeepSORT).
3. Розрахунок відстані S між центрами прямокутників (bounding boxes) у пікселях.
4. Визначення часу t, який пройшов між кадрами (залежно від FPS відео).
5. Застосування масштабного коефіцієнта d для перетворення відстані в метри.
6. Розрахунок швидкості об’єкта.

У коді на Python функція для розрахунку швидкості виглядає так (рис. 3.4):

```python
def compute_speed(prev_box, curr_box, delta_t, pixel_to_meter):
    x1 = (prev_box[0] + prev_box[2]) / 2
    y1 = (prev_box[1] + prev_box[3]) / 2
    x2 = (curr_box[0] + curr_box[2]) / 2
    y2 = (curr_box[1] + curr_box[3]) / 2

    S = ((x2 - x1)**2 + (y2 - y1)**2)**0.5  # Відстань у пікселях
    v = (S * pixel_to_meter) / delta_t      # Швидкість у м/с
    return v * 3.6                          # Переведення в км/год
```

Рис. 3.4 – Функція для розрахунку швидкості

Щоб правильно оцінювати зміну координат об’єкта, необхідно точно співставляти його між кадрами. Для цього була реалізована інтеграція з трекером DeepSORT, який використовує як координатну інформацію, так і візуальні ознаки (appearance features) для забезпечення стабільного відстеження об’єктів навіть при частковому перекритті чи зміні освітлення.

Після виявлення транспортних засобів на відеопотоці та обчислення їхньої швидкості наступним логічним кроком є формування звіту, який містить ключову інформацію про всі виявлені об'єкти. Такий звіт дозволяє зберігати результати аналізу у структурованій формі, проводити подальший аналіз даних, здійснювати моніторинг дорожнього трафіку, а також фіксувати потенційні порушення швидкісного режиму.

Є п’ять основних класи:

- Car – легкові автомобілі.
- Truck – вантажні автомобілі.
- Bus – автобуси.
- Motorcycle – мотоцикли.
- Bicycle – велосипеди.

Ця класифікація здійснюється автоматично на основі виходу детектору об'єктів (YOLOv7), який повертає як координати обмежувальних рамок, так і клас об’єкта.

Для зручності аналізу звіт формується у вигляді таблиці, кожен рядок якої відповідає окремому об’єкту. В таблиці відображаються такі дані:

- Унікальний ідентифікатор об’єкта (ID)
- Клас транспортного засобу (Car, Truck, тощо)
- Час появи в кадрі
- Час зникнення
- Середня швидкість (в км/год)
- Перевищення допустимої швидкості (так/ні)

Нижче наведено приклад коду на Python, який створює подібний звіт у форматі Excel на основі даних розпізнавання.

```
import pandas as pd
from datetime import datetime

# Список словників – імітація отриманих даних з моделі
data = [
    {"id": 1, "class": "Car", "start_time": "12:01:03", "end_time":
"12:01:07", "speed": 58},
    {"id": 2, "class": "Truck", "start_time": "12:01:10", "end_time":
"12:01:16", "speed": 72},
    {"id": 3, "class": "Bicycle", "start_time": "12:01:14", "end_time":
"12:01:20", "speed": 18},
    {"id": 4, "class": "Bus", "start_time": "12:01:25", "end_time":
"12:01:30", "speed": 64},
    {"id": 5, "class": "Motorcycle", "start_time": "12:01:32",
"end_time": "12:01:35", "speed": 85},
]

# Встановлюємо обмеження швидкості за класами
speed_limits = {
    "Car": 60,
    "Truck": 60,
    "Bus": 60,
    "Motorcycle": 70,
    "Bicycle": 25
```

```
}

# Додаємо поле про перевищення швидкості
for item in data:
    limit = speed_limits[item["class"]]
    item["speed_violation"] = "Так" if item["speed"] > limit else "Ні"

# Створюємо DataFrame
df = pd.DataFrame(data)

# Зберігаємо у файл
current_time = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
filename = f"vehicle_speed_report_{current_time}.xlsx"
df.to_excel(filename, index=False)

print(f"Звіт збережено: {filename}")
```

Цей скрипт створює Excel-файл із поточною датою у назві, де зберігаються результати розпізнавання транспортних засобів.

Нижче подано приклад табличної частини сформованого звіту (рис. 3.5).

ID	Клас	Час появи	Час зникнення	Середня швидкість (км/год)	Перевищення швидкості
1	Car	12:01:03	12:01:07	58	Ні
2	Truck	12:01:10	12:01:16	72	Так
3	Bicycle	12:01:14	12:01:20	18	Ні
4	Bus	12:01:25	12:01:30	64	Так
5	Motorcycle	12:01:32	12:01:35	85	Так

Рисунок 3.5 – Приклад звіту

Обмеження швидкості за класами можна встановлювати вручну в залежності від дорожньої ситуації.

Автоматичне формування звіту після розпізнавання транспортних засобів і визначення їхньої швидкості є важливою функціональною частиною системи розпізнавання швидкості. Завдяки цьому забезпечується структуроване зберігання результатів, можливість аналітики та виявлення порушень. Такий

підхід є ефективним інструментом у сфері безпеки дорожнього руху та в міських транспортних інфраструктурах.

ВИСНОВКИ

У результаті виконання роботи було розроблено та реалізовано інформаційну підсистему для автоматизованого розпізнавання транспортних засобів, визначення їхньої швидкості та формування звітів про результати спостереження. Запропоноване рішення охоплює всі основні етапи аналізу відеопотоку та забезпечує точне виявлення і супровід транспортних об’єктів у реальному часі.

На першому етапі реалізовано розпізнавання транспортних засобів з використанням нейронної мережі YOLOv7, яка забезпечила високу точність і швидкодію при виявленні об'єктів транспортної категорії на кожному кадрі відео. Модель успішно ідентифікує п’ять основних класів транспортних засобів: легкові автомобілі, вантажні автомобілі, автобуси, мотоцикли та велосипеди.

Другий етап полягав у трекінгу виявлених об’єктів, який було реалізовано на основі алгоритму DeepSORT. Це дозволило забезпечити стійке відстеження кожного транспортного засобу протягом усього часу його перебування в зоні контролю, навіть за умови часткового перекриття, зміни масштабу або швидкого руху.

У рамках третього етапу здійснено маркування контрольних ліній – умовних меж, які дозволяють фіксувати моменти входу та виходу транспортного засобу з контрольованої ділянки. Встановлення логічних ліній на кадрах є ключовим для визначення часу проходження, необхідного для подальшого розрахунку швидкості.

На четвертому етапі реалізовано алгоритм обчислення швидкості руху транспортних засобів. Зафіксувавши час перетину кожної з контрольних ліній, система розраховує час проходження ділянки та, відповідно, швидкість, з урахуванням масштабу відео (кількість метрів на піксель). Такий підхід дозволяє отримувати досить точні значення швидкості навіть у складних умовах відеозйомки.

П’ятий, завершальний етап передбачає формування звіту, який містить усю необхідну інформацію для подальшого аналізу: унікальний ідентифікатор об’єкта, його тип (клас), середню швидкість руху та факт перевищення допустимого швидкісного режиму. Звіт формується у вигляді структурованої таблиці у форматах Excel або CSV, що дає змогу легко інтегрувати його у бази даних або системи моніторингу дорожнього руху.

Таким чином, розроблена підсистема є повнофункціональним інструментом для інтелектуального аналізу транспортних потоків. Вона може бути використана для інтеграції у систему автоматичного контролю дорожнього руху, виявлення порушень правил, створення транспортної статистики, а також у складі комплексних систем "розумного міста".

СПИСОК ЛІТЕРАТУРИ

1. Jain, A., & Jain, S. (2020). Real-time vehicle speed estimation using YOLO and DeepSORT. IEEE International Conference on Intelligent Transportation.
2. Papageorgiou, C., & Poggio, T. (2000). A trainable system for object detection. International Journal of Computer Vision.
3. Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed.). Pearson.
4. Ghosal, A., & Bandyopadhyay, S. K. (2021). Review on Traffic Surveillance Using Computer Vision. ACM Computing Surveys.
5. Yao, H., Cao, Y., & Wang, Y. (2017). Vehicle speed detection based on video surveillance. Proceedings of the International Conference on Computer Vision Systems.
6. Khatri R., Patil S. A survey on vehicle detection techniques in intelligent transportation systems // Procedia Computer Science. - 2020. - Vol. 167. - С. 2394–2401.
7. Мінаєв О. Ю. Методи обробки та аналізу відеоінформації. - Харків: ХНУРЕ, 2021.
8. Huang G., Liu Z., Van Der Maaten L., Weinberger K. Densely connected convolutional networks (DenseNet) // CVPR. - 2017.
9. Kalman R. E. A New Approach to Linear Filtering and Prediction Problems // Journal of Basic Engineering. - 1960. - Vol. 82. - С. 35–45.
10. Szeliski, R. (2022). Computer Vision: Algorithms and Applications. Springer.
11. Штерц Р.О. АСПЕКТИ СТАБІЛЬНОГО РОЗВИТКУ ЕКОНОМІКИ В УМОВАХ РИНКОВИХ ВІДНОСИН, матеріали XIX Міжнародної науково-практичної конференції. 27 травня 2024 р.
12. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement // arXiv preprint arXiv:1804.02767. – 2018.
13. Wang C. Y., Bochkovskiy A., Liao H. Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors // arXiv preprint arXiv:2207.02696. – 2022.

14. Wojke N., Bewley A., Paulus D. Simple online and realtime tracking with a deep association metric // IEEE International Conference on Image Processing (ICIP). – 2017.
15. Zhang Y., Sun P., Jiang Y., Yu T., Weng F., Yuan Z. ByteTrack: Multi-object tracking by associating every detection box // ECCV. – 2022.
16. Cao J., Pang J., Liu C., Li K., Zhang C., Lin D. Observation-centric SORT: Rethinking SORT for robust multi-object tracking // arXiv:2203.14360. – 2022.
17. Rosebrock A. Deep Learning for Computer Vision with Python. – PyImageSearch, 2018.
18. Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press, 2016.
19. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // NeurIPS. – 2012.
20. Girshick R. Fast R-CNN // Proceedings of the IEEE International Conference on Computer Vision. – 2015.
21. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks // NeurIPS. – 2015.
22. Lin T.-Y., Goyal P., Girshick R., He K., Dollár P. Focal loss for dense object detection // ICCV. – 2017.
23. Bochkovskiy A., Wang C. Y., Liao H. Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection // arXiv:2004.10934. – 2020.
24. OpenCV: Open Source Computer Vision Library [Електронний ресурс]. – Режим доступу: https://opencv.org/
25. GeeksforGeeks. Vehicle Speed Detection using OpenCV and Python [Електронний ресурс]. – 2021.
26. OpenMMLab. MMDetection Framework. – https://github.com/open-mmlab/mmdetection
27. Paszke A., Gross S., Massa F., et al. PyTorch: An imperative style, high-performance deep learning library // NeurIPS. – 2019.

28. TensorFlow. An end-to-end open-source platform for machine learning [Електронний ресурс]. – https://www.tensorflow.org/
29. Tan M., Le Q. EfficientNet: Rethinking model scaling for convolutional neural networks // ICML. – 2019.
30. Rezatofighi H. et al. Generalized Intersection over Union: A metric and a loss for bounding box regression // CVPR. – 2019.
31. Lundberg S. M., Lee S. I. A Unified Approach to Interpreting Model Predictions // NeurIPS. – 2017.
32. Li X., Liu Y., Zha Z., Wang Y. Vehicle detection from drone images using YOLOv5 and EfficientDet // Sensors. – 2021.
33. Tsai D., Hsieh C., Lai S., Lin C. Real-time vehicle speed estimation from traffic surveillance video // Sensors. – 2020.
34. Park S., Jeon J., Kim D. Vision-based vehicle tracking and speed estimation // International Journal of Automotive Technology. – 2020.
35. Єфіменко І. І., Сорокін А. В. Системи технічного зору в транспорті. – К.: НАУ, 2020.
36. Мартинов С. О. Комп’ютерний зір та його застосування в транспортних системах. – К.: Політехніка, 2021.
37. ДСТУ ISO/IEC 30170:2020. Інформаційні технології. Японська мова програмування Python.
38. Сухоруков А. І. Моделі та алгоритми розпізнавання образів. – Харків: ХНУРЕ, 2019.
39. Лавренюк А. І., Барановський О. В. Інтелектуальні системи обробки відеоінформації. – Львів: Видавництво ЛНУ, 2020.
40. Ткаченко О. В. Методи аналізу та обробки зображень у реальному часі. – К.: КНУ, 2022.