

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«КОМП'ЮТЕРНИЙ ЗІР У СИСТЕМАХ АВТОМАТИЧНОЇ
ІДЕНТИФІКАЦІЇ ТА КАТЕГОРИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ»**

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.*

(підпис)

Марк ЖИЛІН
(Ім'я ПРІЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
гр. ІСДМ-61

Керівник:
PhD (доктор філософії)

Рецензент:

Марк ЖИЛІН
(ім'я, ПРІЗВИЩЕ)

Валентина ДАНИЛЬЧЕНКО
(ім'я, ПРІЗВИЩЕ)

Ім'я, ПРІЗВИЩЕ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти магістр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

_____ Каміла СТОРЧАК

« _____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Жиліна Марка Ігоровича

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Комп'ютерний зір у системах автоматичної ідентифікації та категоризації транспортних засобів.

керівник кваліфікаційної роботи Валентина ДАНИЛЬЧЕНКО, PhD

(Ім'я, ПРИЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від « 30 » жовтня 2025 р. №467

2. Строк подання кваліфікаційної роботи « 26 » грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи:

1. завдання на кваліфікаційну роботу студента
2. наукові статті
3. технічна література

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз та огляд предметної області.
2. Постановка задачі та методологія дослідження.
3. Розробка запропонованого алгоритму та засобів реалізації.
4. Реалізація, експериментальні дослідження та результати.

5. Перелік ілюстративного матеріалу: презентація на слайдах

1. Структура згорткової нейронної мережі

2. Архітектура YOLO
3. Діаграми оцінки точності та швидкості роботи методу
4. Результат роботи програми

6. Дата видачі завдання « 30 » жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	30.10.2025	Викон.
2	Написання 1 розділу кваліфікаційної роботи	10.11.2025	Викон.
3	Написання 2 розділу кваліфікаційної роботи	24.11.2025	Викон.
4	Написання 3 розділу кваліфікаційної роботи	29.11.2025	Викон.
5	Висновки, вступ, реферат	28.11.2025	Викон.
6	Перевірка кваліфікаційної роботи на оригінальність тексту	01.12.2025	Викон.
7	Розробка презентації	12.12.2025	Викон.

Здобувач вищої освіти

_____ (підпис)

Марк ЖИЛІН

(Ім'я ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи

_____ (підпис)

Валентина ДАНИЛЬЧЕНКО

(Ім'я ПРІЗВИЩЕ)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій
ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Жилін М.І. до захисту кваліфікаційної роботи
(*прізвище та ініціали*)
за спеціальністю 126 Інформаційні системи та технології
(*код, найменування спеціальності*)
освітньо-професійної програми Інформаційні системи та технології
(*назва*)
на тему: «Комп'ютерний зір у системах автоматичної ідентифікації та категоризації
транспортних засобів»

Кваліфікаційна робота і рецензія додаються.

Директора ННІТ

(*підпис*)

Катерина НЕСТЕРЕНКО

(*Ім'я, ПРІЗВИЩЕ*)

Висновок керівника кваліфікаційної роботи

За час роботи над кваліфікаційною роботою здобувач Жилін Марк Ігорович продемонстрував високий рівень теоретичної та інженерної підготовки, успішно застосувавши сучасні методи Комп'ютерного зору та глибокого навчання для розробки та експериментального дослідження алгоритму ідентифікації та категоризації транспортних засобів.

Він якісно провів аналіз архітектур нейронних мереж (CNN), обґрунтував засоби реалізації та отримав практично значущі результати, виявивши при цьому ініціативність та хист до наукової діяльності. З огляду на глибину опрацювання матеріалу та практичну цінність роботи, Жилін Марк Ігорович заслуговує на оцінку «відмінно» та присвоєння кваліфікації магістр з інформаційних систем та технологій.

Керівник кваліфікаційної роботи

(*підпис*)

Валентина ДАНИЛЬЧЕНКО

(*Ім'я, ПРІЗВИЩЕ*)

« ____ » _____ 2025 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Жилін М.І. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри Інформаційних систем та технологій

(*назва*)

(*підпис*)

Каміла СТОРЧАК

(*Ім'я, ПРІЗВИЩЕ*)

« ____ » _____ 2025 року

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу
на здобуття освітнього ступеня магістра

здобувача вищої освіти Жилін Марк Ігорович

(прізвище, ім'я, по батькові)

на тему «Комп'ютерний зір у системах автоматичної ідентифікації та категоризації транспортних засобів»

Актуальність.

Тема роботи "Комп'ютерний зір у системах автоматичної ідентифікації та категоризації транспортних засобів" є надзвичайно актуальною в умовах прискореного розвитку інтелектуальних транспортних систем та необхідності підвищення безпеки, ефективності та автоматизації дорожнього руху. Застосування глибокого навчання та комп'ютерного зору є критично важливим для створення надійних систем моніторингу, контролю доступу, стягнення плати та обліку транспортних потоків. Дослідження та впровадження високоточних моделей для ідентифікації та категоризації транспортних засобів відповідає сучасним технічним викликам і має значну практичну цінність..

Позитивні сторони.

1. Робота глибоко інтегрує сучасні підходи глибокого навчання (зокрема, архітектури згорткових нейронних мереж) для вирішення складних задач детекції та класифікації транспортних засобів, що демонструє високий інженерний рівень.
2. Досліджується застосування обраних моделей комп'ютерного зору (YOLO, R-CNN) для автоматичної ідентифікації транспортних засобів в умовах реального часу, що підвищує практичну значущість системи.
3. Сформульована та реалізована концептуальна модель системи ідентифікації та категоризації, що є цінним практичним результатом, підтвердженим експериментальними дослідженнями та оцінкою метрик якості.

Недоліки.

1. Недостатньо детально розглянуто питання стійкості запропонованої моделі до несприятливих умов навколишнього середовища.
2. Порівняльний аналіз отриманих результатів обмежений базовими моделями, було б доцільно ширше порівняти продуктивність з найсучаснішими рішеннями у цій галузі.

Відзначені зауваження не впливають на загальну позитивну оцінку магістерської кваліфікаційної роботи

Висновок: *кваліфікаційна робота на здобуття ступеня магістра заслуговує оцінку "відмінно", а здобувач Жилін Марк Ігорович заслуговує присвоєння кваліфікації: магістр з інформаційних систем та технологій.*

Рецензент

науковий ступінь, вчене звання

_____ (підпис)

_____ (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 75 стор., 22 рис., 3 додатки, 34 джерела.

Метою дослідження є розробка ефективного методу для автоматизованої системи виявлення та класифікації транспортних засобів. Ключовою вимогою до системи є її здатність працювати в реальному часі та забезпечувати високу точність результатів навіть за складних умов дорожнього середовища.

Об'єкт дослідження – Алгоритми глибокого навчання та методи комп'ютерного зору у контексті автоматичної обробки візуальної інформації для ідентифікації об'єктів.

Предмет дослідження – Модифікована архітектура згорткової нейронної мережі YOLOv8, методи та технічні рішення, що забезпечують високоточне виявлення та класифікацію транспортних засобів в інтегрованих системах моніторингу дорожнього руху.

У роботі розглянуто актуальність застосування комп'ютерного зору для автоматизації моніторингу дорожнього руху. Проаналізовано переваги та недоліки сучасних архітектур глибокого навчання, що використовуються для детекції об'єктів. Досліджено архітектуру YOLOv8 та запропоновано її модифікацію шляхом впровадження модуля додаткової уваги для покращення виділення важливих ознак. Розроблено метод для інтегрованої системи, здатний працювати в реальному часі. Проведено експериментальне дослідження ефективності та точності запропонованої моделі порівняно з базовими рішеннями.

КЛЮЧОВІ СЛОВА: ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЯ ТРАНСПОРТНИХ ЗАСОБІВ, ГЛИБОКЕ НАВЧАННЯ, YOLOV8, ОБРОБКА ЗОБРАЖЕНЬ, КОМП'ЮТЕРНИЙ ЗІР, МОНІТОРИНГ ТРАНСПОРТУ.

ABSTRACT

The text part of the Master's qualification thesis: 75 pages, 22 figures, 3 appendices, 34 references.

Research Goal the goal of the research is to develop an effective method for an automated system of vehicle detection and classification. A key requirement for the system is its ability to operate in real-time and ensure high accuracy of results even under complex road environment conditions.

Object of Research the object of research is Deep Learning Algorithms and Computer Vision methods in the context of automatic processing of visual information for object identification.

Subject of Research the subject of research is the Modified YOLOv8 Convolutional Neural Network Architecture, along with methods and technical solutions that ensure high-precision vehicle detection and classification in integrated road traffic monitoring systems.

Summary of the Thesis the thesis addresses the relevance of applying Computer Vision for the automation of road traffic monitoring. It analyzes the advantages and disadvantages of modern deep learning architectures used for object detection. The YOLOv8 architecture is investigated, and a modification is proposed by integrating an Attention Module to enhance the extraction of important features. A method for an integrated system capable of operating in real-time has been developed. An experimental study of the efficiency and accuracy of the proposed model has been conducted and compared with baseline solutions.

KEYWORDS: VEHICLE DETECTION AND CLASSIFICATION, DEEP LEARNING, YOLOV8, IMAGE PROCESSING, COMPUTER VISION, TRAFFIC MONITORING.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 МЕТОДОЛОГІЧНІ ТА ТЕОРЕТИЧНІ ОСНОВИ ІДЕНТИФІКАЦІЇ ТА КЛАСИФІКАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ	13
1.1 Аналіз предметної галузі та обґрунтування актуальності дослідження	13
1.2 Огляд методик і підходів до ідентифікації та класифікації транспортних засобів	14
1.2.1 Алгоритм Scale-Invariant Feature Transform	16
1.2.2 Згорткові нейронні мережі	18
1.2.3 Алгоритм YOLO	21
1.3 Аналіз ключових вимог та формування дослідницької задачі	22
Висновки до розділу 1	24
РОЗДІЛ 2 МЕТОДОЛОГІЧНИЙ ПІДХІД ДО ІДЕНТИФІКАЦІЇ ТА КАТЕГОРИЗАЦІЇ АВТОТРАНСПОРТУ	26
2.1 Формування інформаційної бази та огляд алгоритмічних рішень для детекції та розподілу на класи	26
2.2 Опис алгоритму ідентифікації та категорійного розподілу об'єктів .	29
2.3 Методики комп'ютерного зору, що застосовуються для обробки та аналізу візуальної інформації	33
2.4 Методологія підготовки та реалізації процесу детектування й класифікації об'єктів	38
2.5 Оцінка та аналіз ефективності класифікаційних алгоритмів	40
Висновки до розділу 2	44
РОЗДІЛ 3 АРХІТЕКТУРА ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	45
3.1 Деталізація програмно-технологічного стеку та інструментарію реалізації	45
3.2 Імплементация, тренування та валідація моделі з модулем уваги СВАН	47
3.3 Експериментальна верифікація та порівняльний аналіз продуктивності	57
Висновки до розділу 3	62
ВИСНОВКИ	64
ПЕРЕЛІК ПОСИЛАНЬ	66

ДОДАТОК А	70
ДОДАТОК Б	73
ДОДАТОК В	75
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)	78

ВСТУП

Транспортна інфраструктура є фундаментальною опорою сучасного суспільства, гарантуючи безперебійне функціонування економіки, забезпечуючи мобільність громадян та підтримуючи високий рівень безпеки перевезень. Один із пріоритетних напрямків розвитку інтелектуальних транспортних систем (ІТС) полягає в ефективному управлінні та моніторингу транспортних потоків. Це дає змогу істотно мінімізувати затори, оптимізувати екологічні показники міського середовища та підвищити безпеку на автомобільних дорогах. Ключовим елементом для реалізації цих цілей є точне виявлення та класифікація всіх типів транспортних одиниць, що забезпечує збір необхідної емпіричної бази для подальшого аналізу та прийняття стратегічних управлінських рішень.

Актуальність теми зумовлена об'єктивною потребою у створенні надійних та високопродуктивних систем моніторингу дорожнього руху, здатних в автоматичному режимі ідентифікувати види транспортних засобів, відстежувати їхню траєкторію та гарантувати обробку даних у режимі реального часу. Відомо, що традиційні методи візуального аналізу та обробки зображень не завжди є ефективними, особливо в складних умовах (наприклад, при недостатньому освітленні або високій щільності руху). У зв'язку з цим, сучасні підходи на основі технологій глибокого навчання, зокрема згорткові нейронні мережі (CNN), відкривають нові горизонти для значного підвищення точності та стійкості подібних систем.

Мета роботи - розробка вдосконаленого методу автоматизованого виявлення та класифікації транспортних засобів. В основі методу лежить модифікована архітектура YOLOv8, яка має забезпечити підвищену точність та можливість функціонування в реальному часі. Покращення досягається шляхом інтеграції спеціалізованого атеншн-модуля (Attention Module), що дозволяє системі зосередитися на найбільш інформативних ознаках об'єктів навіть за наявності перешкод.

Для досягнення поставленої мети визначено такі дослідницькі завдання:

- Провести критичний огляд та систематичний аналіз поточних наукових підходів і технологій, що використовуються для виявлення та класифікації транспорту у відеопотоках.
- Розробити деталізовану методологію модифікації архітектури YOLOv8 шляхом вбудовування модуля для підвищення якості виявлення малорозмірних та перекритих об'єктів.
- Здійснити практичну імплементацію запропонованого методу виявлення та класифікації на базі модифікованої нейронної мережі.
- Виконати порівняльне оцінювання продуктивності розробленої системи з класичними алгоритмами обробки зображень та стандартними моделями глибокого навчання.
- Верифікувати ефективність запропонованої інтелектуальної системи в різних експлуатаційних умовах: при низькому рівні освітлення, інтенсивному трафіку та швидких змінах дорожньої обстановки.

Об'єкт дослідження - виступає процедура автоматичного ідентифікування та категоризації рухомих транспортних засобів у відеозаписах.

Предметом дослідження є - модифіковані алгоритмічні рішення та методи, що використовуються для автоматизації моніторингу дорожнього руху, спрямовані на збільшення точності та швидкості обробки даних.

У процесі виконання роботи було застосовано комплекс сучасних наукових методів:

Глибоке навчання: Використання згорткових нейронних мереж (CNN) для розпізнавання класів об'єктів на вхідних кадрах.

Моделювання: Застосування та модифікація моделі YOLOv8, інтегрованої зі спеціальним модулем для ефективною локалізації та детальної обробки ознак транспортних одиниць.

Обробка даних: Використання методів попередньої обробки зображень, таких як масштабування, трансформація колірному простору та аугментація (штучне збільшення) даних для підвищення робастності (стійкості) системи.

Відстеження: Застосування алгоритмів трекінгу для безперервного відстеження об'єктів у послідовності кадрів і усунення повторного виявлення.

Експериментальна оцінка: Проведення емпіричних досліджень і валідації розробленої системи на реальних сценаріях дорожнього руху.

Наукова новизна одержаних результатів. Полягає в розробці оригінального методу виявлення та класифікації, який використовує інноваційну модифікацію архітектури YOLOv8 із вбудованим атеншн-механізмом. Завдяки впровадженню цього механізму, запропонований підхід здатний більш ефективно концентруватися на критичних просторових і каналних ознаках. Це забезпечує високу продуктивність системи навіть в екстремальних умовах: при дефіциті освітлення та високій концентрації об'єктів на зображенні.

Практичне значення результатів дослідження. Полягає в можливості безпосереднього впровадження розробленого методу в автоматизовані системи моніторингу трафіку. Ця система пропонує дієвий інструмент для класифікації та ідентифікації транспорту, що сприятиме підвищенню дорожньої безпеки, оптимізації роботи муніципальних транспортних мереж та забезпеченню оперативного аналізу дорожньої обстановки. Такі інтелектуальні рішення є ключовими для сучасного міського відеоспостереження, а також можуть слугувати інструментом стратегічного планування та гнучкого управління транспортними потоками.

Апробація результатів магістерської роботи. Основні положення і результати магістерської роботи доповідались на науково практичних конференціях, що проходили на базі Державного університету інформаційно-комунікаційних технологій.

1 МЕТОДОЛОГІЧНІ ТА ТЕОРЕТИЧНІ ОСНОВИ ІДЕНТИФІКАЦІЇ ТА КЛАСИФІКАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ

1.1 Аналіз предметної галузі та обґрунтування актуальності дослідження

Актуальність дослідницької роботи у сфері комп'ютерного зору, зокрема, з питань виявлення транспортних одиниць, диктується об'єктивною потребою у створенні таких технологічно-програмних комплексів [1]. Важливою складовою, що формує необхідність дослідження в даній галузі, є використання технологій ідентифікації транспортних засобів у безпілотних системах. Це необхідно для виконання критично важливих функцій, спрямованих на забезпечення безпечного переміщення пасажирів або вантажів. На сьогоднішній день сфера автономного транспорту демонструє динамічний розвиток, що, зі свого боку, стимулює прогрес у технологіях розпізнавання транспортних засобів.

Ключова проблема виявлення об'єктів у системах комп'ютерного зору, якими оснащені безпілотні транспортні засоби, полягає у не завжди коректному розпізнаванні транспортних одиниць. Наслідком цієї некоректної детекції є виникнення аварійних ситуацій або, в найгіршому випадку, дорожньо-транспортних пригод.

Потреба у вдосконаленні алгоритмів і систем ідентифікації об'єктів руху зумовлена необхідністю зменшення відсотка хибних спрацьовувань у системах комп'ютерного зору, що, своєю чергою, сприятиме скороченню нещасних випадків на дорогах за участю автономного транспорту. Крім того, висока ресурсомісткість (енерговитратність) поточних алгоритмів і систем ідентифікації є чинником, який уповільнює ширше впровадження цих технологій.

Важливим аргументом на користь актуальності теми дослідження алгоритмів виявлення транспортних засобів є їхнє використання у системах інтелектуального моніторингу дорожнього руху. Такі системи, що використовують технології комп'ютерного зору та відповідні алгоритми, забезпечують більш ефективно управління транспортними потоками у зонах їхнього застосування [2, 3]. Також слід

підкреслити, що впровадження алгоритмів детекції в системи моніторингу відкриває шлях до оптимізації транспортних потоків. Підвищення оптимізованості транспортної мережі позитивно впливає на загальний стан інфраструктури та сприяє зменшенню ймовірності утворення заторів на автошляхах, що має позитивний вплив на різні аспекти суспільного життя [4]. Додатково, актуальність досліджень у сфері комп'ютерного зору, зокрема, в алгоритмах виявлення транспорту, продиктована потребою посилити загальний стан безпеки.

При експлуатації систем моніторингу дорожнього руху також нерідко спостерігається певний відсоток некоректно виявлених об'єктів, що у деяких випадках може призводити до аварій. Для мінімізації відсотка невірної детекції необхідне вдосконалення алгоритмів у їхніх найбільш проблемних аспектах, а саме – у підвищенні енергоефективності. Це обумовлено тим, що для більш коректного розпізнавання транспортних засобів, особливо в режимі реального часу (відеопотоці), потрібні значні обчислювальні потужності. Існуючі алгоритми, що є високоефективними з точки зору використання обчислювальних ресурсів, часто демонструють нижчий відсоток коректної ідентифікації, що зумовлено певною недосконалістю сучасних методів. На противагу цьому, високоточні за критерієм коректності виявлення алгоритми є ресурсомісткими і, відповідно, низько ефективними з точки зору споживання ресурсів.

1.2 Огляд методик і підходів до ідентифікації та класифікації транспортних засобів

У сучасному методологічному та методичному арсеналі в галузі детекції об'єктів, зокрема транспортних одиниць, існує значна кількість різноманітних моделей, методів та алгоритмів. Одним із класичних підходів у комп'ютерному зорі, який традиційно використовується для розв'язання задачі виявлення транспорту, є метод Гістограми орієнтованих градієнтів (HOG) (Histogram of Oriented Gradients). Метод HOG дозволяє ефективно виділяти інформацію, що стосується контуру та форми розпізнаваних об'єктів [13].

Роботу алгоритму HOG можна розділити на кілька послідовних етапів.

Обчислення градієнтів: На початковій стадії вхідні дані підлягають перетворенню у сітку клітин. Далі для кожної клітини виконується розрахунок градієнта інтенсивності — це векторна величина, що визначає напрямок та амплітуду найбільш стрімкої зміни інтенсивності.

Формування гістограм орієнтації: На наступному кроці для кожної окремої клітини будується гістограма, яка відображає розподіл орієнтацій градієнтів. Кожен стовпчик у цій гістограмі відповідає певному діапазону кутів орієнтації градієнтів. Таким чином отримується первинний вектор ознак, який описує розподіл градієнтів у межах конкретної клітини.

Блокова нормалізація: З метою підвищення інваріантності до варіацій освітлення та контрасту, отримані вектори ознак підлягають нормалізації. Цей процес здійснюється шляхом блокової нормалізації, при якій вектори ознак із сусідніх клітин об'єднуються у так звані блоки, і для кожного такого блоку обчислюється норма.

Створення фінального дескриптора: Завершальний етап полягає у конкатенації (злитті) всіх нормалізованих векторів ознак, отриманих від усіх блоків. Результатом є кінцевий дескриптор HOG для вхідних даних.

Висока ефективність методу Histogram of Oriented Gradients (HOG) пояснюється його здатністю забезпечувати інваріантність щодо змінних умов освітлення. Ця властивість досягається завдяки механізму нормалізації, який робить кінцевий HOG-дескриптор менш чутливим до коливань яскравості.

Крім того, HOG характеризується стійкістю до геометричних спотворень, що гарантує його надійність при зміні масштабу, ротації або незначних деформаціях вхідних даних.

Ефективність HOG також частково обумовлена обчислювальною простотою закладеного в нього алгоритму. Метод є відносно нескладним і може бути швидко реалізований на сучасному комп'ютерному обладнанні, що робить його широко доступним для практичного застосування. Більш того, HOG має високу

дискримінаційну здатність, тобто він дозволяє успішно ідентифікувати об'єкти, що належать до різних класів [14].

Серед основних переваг HOG можна виділити легкість для розуміння та імплементації, а також підтверджену ефективність у різноманітних завданнях, зокрема, у розпізнаванні транспортних засобів. Ще однією значною перевагою є його інваріантність до численних типів змін у вхідних даних, що забезпечує його надійну роботу при аналізі реальних зображень.

Попри численні переваги, метод HOG також має певні суттєві обмеження. Зокрема, він виявляє значну чутливість до шуму у вхідних даних. Наявність шуму може негативно вплинути на його загальну продуктивність, особливо у випадках, коли зображення є низькоякісними або зашумленими. Іншим недоліком HOG є його низька результативність при обробці об'єктів із надто складною внутрішньою текстурою, де згенерований дескриптор може не забезпечити необхідний рівень точності для надійного розпізнавання.

Таким чином, метод HOG залишається надійним і потужним інструментом для завдань, пов'язаних із детекцією об'єктів, завдяки своїй алгоритмічній простоті, обчислювальній ефективності та високій стійкості до різноманітних змін вхідних даних. Проте, його практичне використання вимагає обов'язкового врахування його підвищеної чутливості до шумових ефектів та певних обмежень при аналізі об'єктів зі складною текстурою.

1.2.1 Алгоритм Scale-Invariant Feature Transform

Серед інших підходів до ідентифікації транспортних засобів значною популярністю користується метод Scale-Invariant Feature Transform (SIFT). Цей алгоритм був розроблений для виявлення та детального опису локальних ключових ознак на зображеннях. Основна перевага SIFT полягає в його інваріантності щодо зміни масштабу, кута обертання, рівня освітленості та певних видів афінних перетворень [15–17].

Принцип роботи SIFT включає декілька послідовних ключових фаз:

Визначення екстремумів у масштабно-просторовому поданні: На першій фазі відбувається аналіз зображення на різних масштабах за допомогою масштабно-інваріантної піраміди. Для цього застосовується оператор різниці гаусових фільтрів (DoG), який дає змогу виявити локальні екстремуми як у просторі координат, так і у просторі масштабу.

Уточнення локалізації екстремумів: Далі проводиться уточнення положення знайдених точок шляхом інтерполяції. У цей же час відбувається відсіювання екстремумів, які характеризуються низьким контрастом або неточною локалізацією.

Призначення орієнтації ключовим точкам: Для кожної виявленої ключової точки обчислюється її основна орієнтація. Цей процес включає побудову гістограми градієнтів у невеликій локальній ділянці навколо точки. Домінантна орієнтація визначається за найвищим піком гістограми.

Створення дескриптора: Завершальний етап – формування вектору ознак (дескриптора), який описує розподіл градієнтів у локальній околиці ключової точки. Дескриптор орієнтується відносно визначеної домінантної орієнтації, що забезпечує його інваріантність щодо обертання зображення.

Переваги та обмеження SIFT

До основних позитивних характеристик SIFT слід віднести його незалежність від масштабу, обертання та змін освітленості, а також високу повторюваність ключових точок. Це означає, що точки, ідентифіковані на різних зображеннях одного об'єкта, матимуть подібні описові вектори. Додатковою перевагою є його висока продуктивність, оскільки архітектура SIFT підтримує паралельні обчислення.

Водночас, метод SIFT має низку недоліків. По-перше, він вирізняється значною обчислювальною складністю – для опрацювання великих масивів даних або забезпечення роботи в режимі реального часу потрібні потужні обчислювальні ресурси. По-друге, алгоритм чутливий до наявності шуму, що може спричинити генерацію помилкових ключових точок на зашумлених зображеннях. Ще одним суттєвим обмеженням SIFT є його обмежена стійкість до значних геометричних

деформацій об'єктів, що знижує його ефективність при роботі з об'єктами, які зазнали сильних трансформацій.

Таким чином, SIFT визнаний надійним підходом для завдань виявлення та опису локальних ознак завдяки його стійкості до масштабних та орієнтаційних змін, а також змін освітленості. Однак, його впровадження вимагає обов'язкового врахування обчислювальних обмежень та його чутливості до шумів у несприятливих умовах експлуатації.

1.2.2 Згорткові нейронні мережі

Серед сучасних підходів до ідентифікації об'єктів широко відомим є клас алгоритмів, що отримав назву згорткові нейронні мережі (Convolutional Neural Networks, CNN). CNN являють собою особливий різновид штучних нейронних мереж, які досягли виняткових успіхів у галузі комп'ютерного зору, зокрема, у детекції об'єктів і, відповідно, виявленні транспортних засобів.

Концепція згорткових нейронних мереж була інспірована біологічними процесами зору, завдяки чому вони є ідеально пристосованими для обробки даних, що мають сігчасту структуру, таких як відеопотік, окремі зображення або їхні послідовності.

Алгоритм роботи CNN має таку структуру:

Згорткові шари (Convolutional Layers): Це ключовий будівельний елемент архітектури CNN. Вони використовують фільтри (ядра), які застосовуються до вхідних даних для виділення характерних ознак, таких як контури, кути та текстурні особливості.

Шари підвибірки (Pooling Layers): Ці шари відповідають за зменшення просторової розмірності обробленого зображення, одночасно зберігаючи найбільш важливу інформацію.

Повністю з'єднані шари (Fully Connected Layers): На завершальній стадії ці шари класифікують виділені ознаки та формують кінцевий результат [18].

Згорткові нейронні мережі (CNN) являють собою потужну технологію для розпізнавання та категоризації зображень, що робить їх надзвичайно придатними для вирішення завдань, пов'язаних із детекцією транспортних засобів. Наша мета – провести класифікацію зображень, що містять транспорт (наприклад, легкові авто, вантажівки, мікроавтобуси чи велосипеди).

Початкова фаза процесу (див. Рис. 1.1) передбачає подачу зображення (наприклад, автомобіля) на вхід мережі. З точки зору комп'ютера, це є не що інше, як масив значень пікселів, що формують візуальну картину. Це зображення послідовно проходить через кілька етапів глибокої обробки.

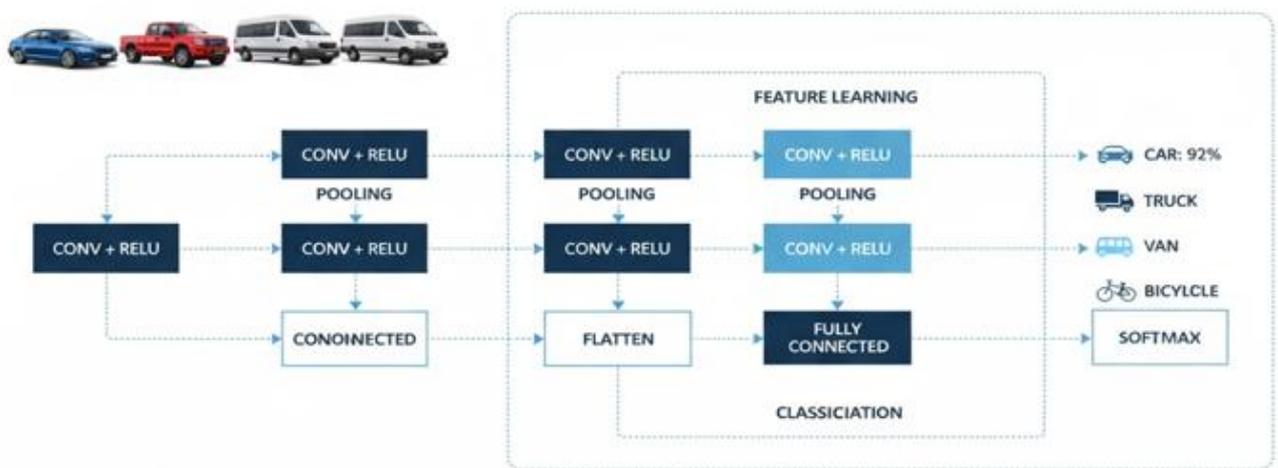


Рис. 1.1 – Загальна архітектура згорткової нейронної мережі для класифікації

На початковій фазі обробки даних, мережа застосовує операцію згортки (convolution), в якій відбувається виділення фундаментальних ознак зображення, таких як контури об'єктів, текстурні елементи або колірні переходи. Цей процес дає змогу нейронній мережі «розпізнати» основні компоненти, що формують об'єкт — наприклад, обрис кузова або специфічну форму колеса. Після цього застосовується функція активації ReLU (Rectified Linear Unit), завданням якої є активація лише позитивних значень і відкидання негативних. Це суттєво прискорює процес навчання та підвищує загальну ефективність роботи мережі.

Наступним кроком є застосування шару підвибірки (pooling), який відповідає за скорочення просторового розміру представлення зображення. Це дозволяє

зберегти виключно значущі ознаки, одночасно забезпечуючи зниження обчислювальних витрат. Наприклад, мережа може зберегти тільки визначальні характеристики об'єкта, такі як форма кузова або кількість осей, ігноруючи другорядні деталі.

Ці послідовні етапи повторюються багаторазово, і з кожним повторенням мережа екстрагує все більш складні та абстрактні ознаки. Якщо на перших ітераціях фіксуються лише базові контури, то на наступних етапах архітектура починає ідентифікувати більш специфічні елементи, такі як колеса чи дверні отвори автомобіля.

Коли вся візуальна інформація оброблена, наступним обов'язковим кроком є площинення (flattening) – процес перетворення багатовимірних даних на єдиний вектор. Цей вектор далі подається на вхід повнозв'язаного шару (fully connected layer), де мережа здійснює комбінування всіх отриманих ознак для формування фінального прогнозу.

Завершальна стадія передбачає застосування функції Softmax, яка конвертує вихідні числові дані на ймовірності належності до кожного з визначених класів, наприклад, «автомобіль», «вантажівка», «мікроавтобус», «велосипед» тощо. Мережа обирає клас із найбільшою ймовірністю, і він стає її остаточним результатом.

Наведемо приклад: якщо нейронна мережа отримує зображення автомобіля, вона спочатку ідентифікує базові візуальні атрибути (контури, колеса). На глибших рівнях вона навчається розпізнавати унікальні деталі, характерні саме для класу «автомобіль». В результаті, після завершення всіх етапів обробки, мережа видасть високу ймовірність належності до класу «автомобіль», і цей клас буде затверджено як результат.

Такий ієрархічний підхід дозволяє CNN не тільки ефективно класифікувати транспортні засоби, але й демонструвати високу стійкість до різноманітних зовнішніх факторів, як-от зміни в освітленні або незначні деформації об'єкта на зображенні. Застосування подібних методик є критично важливим для систем

автоматизованого моніторингу дорожнього руху, де точність ідентифікації та класифікації різних типів транспорту є пріоритетом.

Ефективність CNN підтверджується, зокрема, їхньою інваріантністю до зсуву – CNN здатні виявляти ознаки об'єктів незалежно від їхнього розташування в межах вхідних даних. Крім того, ефективність CNN при роботі з візуальними даними (зображеннями та відео) підтверджується використанням локального виявлення ознак: фільтри CNN дозволяють виділяти локальні, ключові характеристики, що є вирішальним етапом для успішного розпізнавання, наприклад, транспортних засобів. Також висока продуктивність CNN доводиться тим фактом, що в них реалізована ієрархічна обробка даних: інформація послідовно обробляється різними шарами, переходячи від простих ознак до все більш абстрактних представлень [19].

Ще однією важливою перевагою CNN є висока точність: ці мережі здатні демонструвати видатні результати за метрикою точності у більшості завдань комп'ютерного зору, включаючи детекцію та класифікацію різних типів транспортних об'єктів.

1.2.3 Алгоритм YOLO

YOLO (You Only Look Once) – це спеціалізований алгоритм, який базується на принципах глибокого навчання і був розроблений для ефективного виявлення об'єктів як на статичних зображеннях, так і у відеопотоках. Методика роботи YOLO полягає у наступному: на початковій фазі вхідні дані розділяються на рівномірну сітку. Кожна клітина цієї сітки несе зону відповідальності за прогнозування тих об'єктів, центри яких знаходяться всередині цієї клітини. Далі відбувається процес прогнозування граничних рамок (bounding boxes), де для кожної клітини мережа генерує декілька потенційних рамок [20-21].

Серед ключових переваг YOLO можна виділити його високу швидкість роботи. Крім того, алгоритм YOLO є відносно простим у використанні та розумінні його архітектури. Ще однією значною перевагою є цілісне сприйняття вхідних

даних: YOLO обробляє зображення як єдине ціле, що дозволяє йому враховувати контекст і успішно виявляти об'єкти, що частково перекриті. Проте, алгоритм YOLO має і певні недоліки, зокрема – знижену точність при виявленні дрібних об'єктів та чутливість до некоректного співвідношення сторін (аспектрального співвідношення).

1.3 Аналіз ключових вимог та формування дослідницької задачі

Об'єктом даного наукового дослідження є інтелектуальна система, призначена для виявлення та класифікації транспортних засобів, що функціонує на базі алгоритмів глибокого навчання, зокрема, з використанням згорткових нейронних мереж. У контексті роботи до категорії транспортних засобів відносяться легкові автомобілі, автобуси, вантажний транспорт, мотоцикли та інші види об'єктів, які підлягають детекції на основі вхідних візуальних даних. Основний акцент у дослідженні приділяється продуктивності та надійності системи в умовах різного типу трафіку, включно з високою інтенсивністю руху, зйомкою з різних кутів та роботою у складному дорожньому середовищі.

Модифікація та вдосконалення алгоритму

Дослідницька частина роботи ґрунтується на застосуванні модифікованої версії алгоритму YOLOv8, який було доопрацьовано з метою підвищення точності розпізнавання та загальної ефективності ідентифікації об'єктів. Ця архітектура інтегрує традиційні принципи YOLO з архітектурними покращеннями, зокрема, із застосуванням механізмів підвищеної уваги (attention modules). Це впровадження призначене для посилення фокусування нейронної мережі на найбільш інформативних ознаках об'єктів.

З огляду на те, що якість результатів функціонування алгоритмів глибокого навчання критично залежить від параметрів освітлення та характеристик вхідних зображень, передбачається ретельний аналіз продуктивності розробленої моделі в різноманітних умовах. Це включає денне освітлення, ситуації зниженої видимості, вплив несприятливих погодних чинників та наявність часткового перекриття

об'єктів. Особлива увага приділяється сценаріям, які максимально відповідають реальним умовам зйомки, що здійснюється вуличними відеокамерами [23].

Формулювання ключових вимог до системи

Критично важливі вимоги до розроблюваної системи охоплюють кілька принципових аспектів.

Точність детекції: Це є провідною вимогою, оскільки система має надійно ідентифікувати транспортні засоби різних категорій, мінімізуючи кількість помилок – як хибних позитивних (помилкове розпізнавання об'єкта), так і хибних негативних (ігнорування наявного об'єкта).

Стійкість до зовнішніх умов: Система повинна демонструвати високу робастність (стійкість) до зовнішніх факторів, включаючи зміни рівня освітленості, фрагментарне перекриття об'єктів, варіації ракурсів зйомки та вплив метеорологічних умов.

Постановка основного завдання дослідження

Основним завданням даного дослідження є створення вдосконаленої моделі YOLOv8, яка здатна поєднати високу швидкість обробки даних із підвищеною точністю виявлення та класифікації транспортних засобів. Для досягнення цієї мети модель була навчена на кількох репрезентативних датасетах, зокрема наборі даних "Vehicle Detection: 8 Classes" із платформи Kaggle. Цей набір містить анотовані зображення різноманітних типів транспорту, відзняті в різних умовах. Використання диверсифікованих наборів даних дозволяє моделі набути здатності до адаптації до реальних сценаріїв, включаючи як складне освітлення, так і високу щільність об'єктів у кадрі [23, 24].

Для валідації узагальнюючої здатності моделі було проведено тестування на незалежних наборах даних. Це дало змогу об'єктивно оцінити її точність, стійкість і продуктивність у різних умовах експлуатації. Експериментальна частина також включала оптимізацію метрик оцінювання для забезпечення детальної та всебічної оцінки ефективності системи.

Підсумовуючи, кінцевою метою даного дослідження є розробка високоточного та енергоефективного алгоритму для виявлення та класифікації

транспорту, заснованого на архітектурі глибоких нейронних мереж. Запропонована система вирізняється комплексом переваг: швидкість, адаптивність та висока точність, що робить її ефективним рішенням для інтелектуального моніторингу дорожнього руху в реальних експлуатаційних умовах.

Висновки до розділу 1

На підставі проведеного аналізу теоретичних та методичних засад ідентифікації транспортних засобів, сформульовані такі ключові результати та висновки:

Проведено ґрунтовний аналіз предметної області, що охоплює сферу комп'ютерного зору та глибокого навчання, а також здійснено критичне оцінювання сучасних підходів і методів, що використовуються для розв'язання задач детекції та категоризації транспортних об'єктів.

Детально розглянуто та проаналізовано традиційні та нейромережеві методи виявлення транспортних засобів, включаючи алгоритми HOG (оцінено його простоту та чутливість до шуму) та SIFT (визначено інваріантність, але обчислювальну складність). Особливу увагу приділено архітектурі та принципам функціонування згорткових нейронних мереж (CNN) як основі для сучасних рішень, а також особливостям алгоритму YOLO як бази для подальшої модифікації.

Ідентифіковано та обґрунтовано ключові операційні вимоги до розроблюваної системи ідентифікації транспортних засобів. До них віднесено: високу точність (точність детекції), здатність ефективно працювати в режимі реального часу, а також надійну стійкість (робастність) до несприятливих зовнішніх чинників – зокрема, до варіацій освітлення, складних погодних умов та часткового перекриття об'єктів у кадрі.

Сформульовано та чітко поставлено дослідницьку задачу, яка полягає у створенні та впровадженні високопродуктивного та енергоефективного методу виявлення і класифікації транспортних засобів на основі модифікованої

нейромережевої архітектури, що має забезпечити оптимальний баланс між швидкістю та точністю.

2 МЕТОДОЛОГІЧНИЙ ПІДХІД ДО ІДЕНТИФІКАЦІЇ ТА КАТЕГОРИЗАЦІЇ АВТОТРАНСПОРТУ

2.1 Формування інформаційної бази та огляд алгоритмічних рішень для детекції та розподілу на класи

Для успішної реалізації методики детектування та категорійного розподілу автотранспортних засобів критично важливо сформувати високоякісний масив даних. Цей набір має гарантувати ефективність процесу навчання та наступного тестування розроблених алгоритмів. Обов'язковою умовою є те, що ці інформаційні одиниці повинні адекватно відображати різноманітні ситуації дорожнього руху, включаючи широкий спектр типів транспорту, мінливість умов освітлення та нестандартні або ускладнені дорожні сценарії.

Основним первинним джерелом необхідної інформації слугують відеоматеріали, що фіксують дорожній рух. Їх отримують із камер відеоспостереження, які спеціально монтують над проїжджою частиною. Розташування камер зазвичай варіюється: від високого ракурсу (що імітує «погляд зверху») до горизонтальної перспективи, типової для розміщення на транспортних розв'язках. Такий підхід забезпечує охоплення значного діапазону ситуацій: від прямолінійних ділянок автомагістралей до багатосмугових та заплутаних перехресть.

Невід'ємною складовою підготовки даних є збирання матеріалів, знятих за різних рівнів освітленості. Для досягнення цієї мети використовуються записи, зроблені як вдень, так і у період сутінків чи вночі. Умови темного часу доби вимагають особливої уваги до фар транспортних засобів, оскільки вони можуть бути використані алгоритмами обробки як важливі маркери чи ідентифікаційні ознаки. Додатково до цього, обов'язково слід включити відео, які фіксують рух за ускладнених кліматичних умов, як-от сильний дощ, густий туман або снігопад.

Окрім безпосереднього запису з індивідуальних камер, доцільно застосовувати загальнодоступні й відкриті інформаційні масиви, такі як СОСО

(Common Objects in Context) або KITTI Vision Benchmark Suite. Ці стандартизовані бази даних вже містять проведену розмітку об'єктів, зокрема транспортних засобів, з точно визначеними координатами граничних рамок (bounding boxes) та призначеними класами (наприклад, легковий автомобіль, вантажний транспорт, автобус, двоколісний транспорт тощо). Використання таких попередньо маркованих даних спрощує процедуру початкового навчання моделей і дозволяє забезпечити підвищену якість категорійного розпізнавання [25].

Підготовлений інформаційний пул повинен містити зображення та відеофайли з високою роздільною здатністю, що є запорукою достатньої деталізації розпізнаваних об'єктів. Першочергова увага має бути приділена процесу маркування даних: для кожного виявленого транспортного засобу необхідно чітко зафіксувати його категорію (клас) та позиційні координати в межах кадру. Ця скрупульозна робота дозволяє успішно тренувати моделі глибокого навчання, зокрема архітектуру YOLO, забезпечуючи винятково високий показник точності розпізнавання.

У контексті детектування та класифікації об'єктів виділяють два фундаментальні методологічні напрямки. Перший базується на використанні нейронних мереж (НМ), які використовують механізм зворотного поширення помилок (backpropagation) у парадигмі "від входу до виходу" (end-to-end). Найбільш поширеними у цій групі є згорткові нейронні мережі (CNN). Другий напрямок включає традиційніші методи, що не використовують НМ, і їхня робота ґрунтується виключно на класичних операціях обробки зображень або задіюють спрощені алгоритми машинного навчання, як-от опорні вектори (Support Vector Machines, SVM).

Розглядаючи еволюційний шлях систем ідентифікації та категорійного розподілу транспорту, варто зазначити, що початкові вагомні результати були досягнуті у сфері підрахунку кількості транспортних одиниць. Ранні, відносно прості системи, які функціонували на засадах комп'ютерного зору, застосовували техінку відокремлення переднього плану (де знаходився рухомий об'єкт) від статичного фону. Найбільш успішні зразки цих підходів залежали від оптимальної

якості зйомки, яка мала забезпечувати достатню кількість пікселів для формування зображення транспортних засобів. Значна частина ранніх методик використовувала поняття оптичного потоку та застосування віртуальних ліній перетину для обліку проїжджаючого транспорту. Однак, великогабаритні одиниці, такі як вантажівки з причепами, часто спричиняли спотворення у підрахунках, оскільки більшість тодішніх підходів не могли гарантувати достатньої точності для таких випадків.

Ще однією істотною перешкодою було те, що практично всі розроблені методи демонстрували низький рівень точності підрахунку в умовах недостатнього освітлення, зокрема у сутінках, за похмурої погоди або вночі. У цьому контексті лише поодинокі рішення виявилися успішними. Втім, залишається загально визнаним факт, що точність ідентифікації та класифікації суттєво знижується за умов обмеженої освітленості, особливо у темний час доби. Крім того, ранні методології страждали від неможливості обробляти інформацію в режимі реального часу. Це було серйозним недоліком, оскільки відсутність оперативної обробки робила такі системи практично непридатними для повноцінного моніторингу дорожнього трафіку.

До 2010 року практично жодна дослідницька робота не могла забезпечити класифікацію транспортних засобів у реальному часі, використовуючи при цьому середньостатистичні обчислювальні потужності. Проте, із стрімким розвитком нейронних мереж, сьогодні досягнуто високих показників точності у категорійному розпізнаванні та обліку транспорту, які функціонують в оперативному режимі навіть із залученням помірних обчислювальних ресурсів. Ключовою передумовою для досягнення таких результатів є забезпечення якісного фронтального огляду транспортного засобу та захоплення достатньої кількості пікселів, що є легкодосяжним при встановленні камер безпосередньо над дорогою. Проте, ця висока ефективність швидко нівелюється за несприятливих світлових умов, таких як сутінки, дощ або значна хмарність.

У контексті нічного моніторингу дорожнього трафіку нейронні мережі зазвичай не застосовуються. Натомість, альтернативні методики, які використовують виявлення світла фар, успішно виконують функцію підрахунку

транспортних засобів. Точне категорійне розпізнавання в нічний час може бути забезпечене лише за умови відмінного освітлення ділянки дороги. Такі ідеальні умови легко створюються штучно, наприклад, у дорожніх тунелях, де камери можуть бути розташовані безпосередньо над смугами руху для забезпечення ефективної ідентифікації та класифікації. За умов високої якості зйомки ці системи також спроможні зчитувати номерні знаки автомобілів на в'їзді та виїзді з добре освітлених зон.

2.2 Опис алгоритму ідентифікації та категорійного розподілу об'єктів

Процедура детектування об'єктів шляхом аналізу відеопотоку та статичних зображень може бути реалізована кількома методами. Основний стратегічний підхід полягає у виокремленні рухомих елементів через застосування масок, подальшому визначенні їхніх контурів і створенні граничних рамок (bounding boxes). Цей метод ґрунтується на експериментальних дослідженнях з оптимізації параметрів масок із обов'язковим урахуванням факторів зовнішнього середовища, в якому знаходяться ці об'єкти.

У рамках розглянутої методології ідентифікації автотранспорту пріоритетно виділяються лише ті об'єкти, що демонструють рух (оскільки камери фіксуються у стаціонарному положенні). Об'єкти, що відповідають заданим розмірним критеріям, маскуються у формі "областей" (blobs), кожна з яких отримує свій центроїд і обмежувальну рамку. Також передбачено механізм обліку цих виявлених елементів через фіксацію моменту, коли центроїд об'єкта перетинає попередньо визначену лінію. З метою підвищення обчислювальної ефективності було визначено область інтересу (Region of Interest, ROI), що обмежувала лише ту ділянку кадру, яка підлягала аналізу. ROI була візуалізована у вигляді тонкого прямокутника (рис. 2.1).

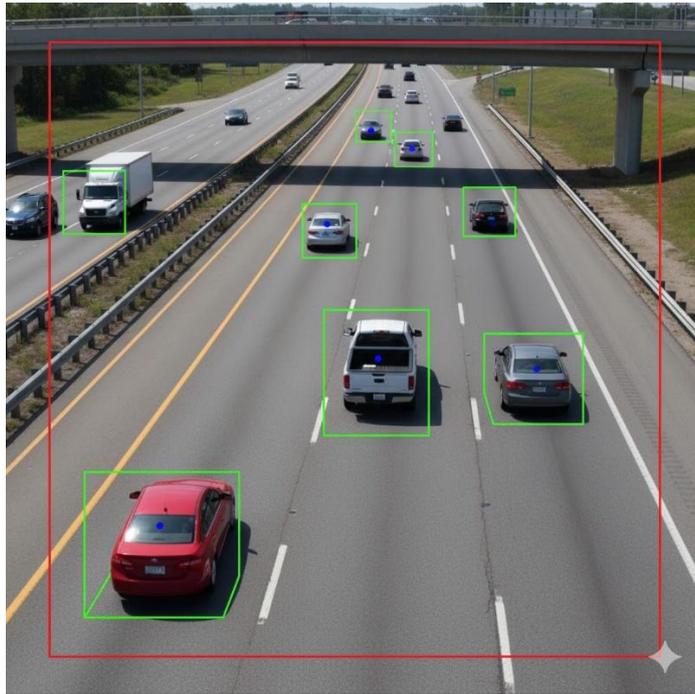


Рис. 2.1 Виділення області інтересу (ROI) для аналізу транспортного потоку

Інші підходи, наприклад, метод k -найближчих сусідів (k -Nearest Neighbors), також можуть бути використані для виокремлення об'єктів. Однак вони характеризуються суттєво вищими обчислювальними витратами, що робить їхнє застосування для детектування транспорту в режимі реального часу неможливим.

Оптимальним рішенням для досягнення балансу між оперативністю та точністю функціонування системи є алгоритм MOG2. Ключовий етап цього підходу – це процедура віднімання фону (Background Subtraction, BS), яка дає змогу сформувати маску переднього плану, тим самим ізолюючи рухомі об'єкти. Для реалізації BS застосовується метод "Суміші Гауссіанів" (Mixture of Gaussians), який визнано одним із найбільш швидких та точних для подібних завдань. Маска переднього плану представлений у вигляді окремого зображення, що ілюструє виділення рухомих об'єктів (рис. 2.2).



Рис. 2.2 Маска видалення фону (результат застосування методу Mixture of Gaussians)

На жаль, методології виявлення об'єктів, що не базуються на нейронних мережах, виявляють низку істотних недоліків, які стали очевидними в процесі експериментів. Оскільки цей процес здійснює виключно інтерпретацію руху, визначаючи відмінності від фонового зображення та враховуючи розміри об'єктів, проблемні ситуації виникають у випадках, коли транспортні засоби частково або повністю перекривають один одного. У таких сценаріях програмне забезпечення ідентифікує перекриті об'єкти як єдине ціле, замість того щоб розпізнати їх як дві окремі одиниці. Це особливо помітно в ситуаціях, коли більш габаритний транспорт, наприклад, вантажний автомобіль, закриває легкове авто, що рухається позаду [22].

Можливим шляхом вирішення цієї складності є використання підвищеного горизонтального ракурсу зйомки або повного вигляду згори (bird's eye view). Однак такий підхід значно обмежує універсальність та практичність подібних рішень, оскільки ідеально функціонуюча система повинна зберігати свою ефективність незалежно від кута розміщення камери. Крім того, аналізована система стикається з іншими функціональними обмеженнями: вона демонструє найкращі результати

виключно на двосмугових дорогах із прямими ділянками. Оптимальна ж система мала б ефективно опрацьовувати транспортні перехрестя, багатосмугові магістралі та криволінійні ділянки, зберігаючи при цьому незалежність від кута зйомки. Проблему накладання транспортних засобів ілюструє (рис. 2.3).

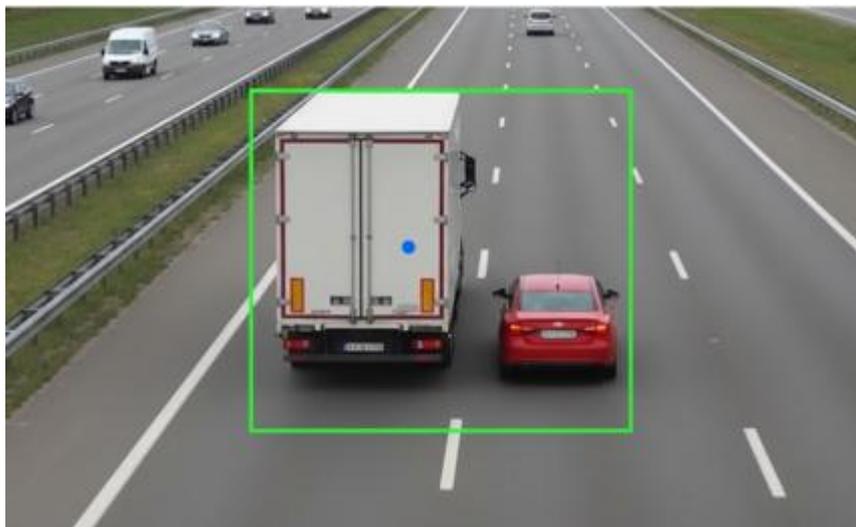


Рис. 2.3 Маска з накладанням об'єктів (ілюстрація проблеми оклюзії)

Ще однією значною перепорою є умови низького рівня освітлення. У таких ситуаціях істотно зростає рівень шуму, через що процедура віднімання фону стає непродуктивною, і система припиняє коректну роботу. Аналогічні труднощі виникають і у випадку надмірної експозиції, коли камера не може правильно інтерпретувати передній план через сильне відбиття світлових променів.

Остання виявлена проблема пов'язана зі зміною площі пікселів об'єкта, що виникає при наближенні транспортного засобу до знімального пристрою. У результаті цієї зміни розмір об'єкта зростає, що спричиняє суттєве зміщення позиції його центроїда. Це, своєю чергою, призводить до неточних результатів підрахунку, особливо коли великі транспортні засоби, наприклад вантажівки, займають надмірно велику піксельну площу. Такі сценарії появи неточних результатів також можуть бути візуально зафіксовані дивись (рис. 2.4).



Рис. 2.4 Приклад причини появи неточних результатів (ілюстрація проблеми зміни розміру об'єкта та зміщення центроїда)

Що стосується фінальних результатів, цей базовий методологічний підхід є досить точним на короткій дистанції, проте при збільшенні тривалості роботи програми похибки неминуче накопичуються внаслідок згаданих проблем перекриття та динамічної зміни розмірів розпізнаваних об'єктів.

2.3 Методики комп'ютерного зору, що застосовуються для обробки та аналізу візуальної інформації

Сучасні системи комп'ютерного зору для оперативного детектування та класифікації об'єктів повсюдно використовують високоефективний алгоритм, відомий як YOLO (You Only Look Once). До моменту його презентації Редмоном у 2015 році, домінантними методами у сфері виявлення об'єктів були підходи типу R-CNN. Однак, ці попередні рішення, хоча й забезпечували високу точність, характеризувалися недостатньою продуктивністю, що робило їх непридатними для ефективного застосування у завданнях, що вимагають обробки в реальному часі [25].

Завдяки своїй винятковій оперативності та високій прецизійності результатів, YOLO здобув значне визнання. Він став провідним інструментом для широкого спектра прикладних завдань, включаючи автоматичне виявлення транспортних засобів, розпізнавання тварин та моніторинг порушень безпеки. Принцип роботи

системи YOLO полягає у послідовній обробці вхідного зображення (як ілюструє рис. 2.5): спочатку відбувається його масштабування до необхідного розміру, після чого зображення проходить через єдину згорткову нейронну мережу. На фінальному етапі, результати фільтруються відповідно до рівня довіри (або оцінки ймовірності) моделі, що забезпечує швидке і точне детектування.

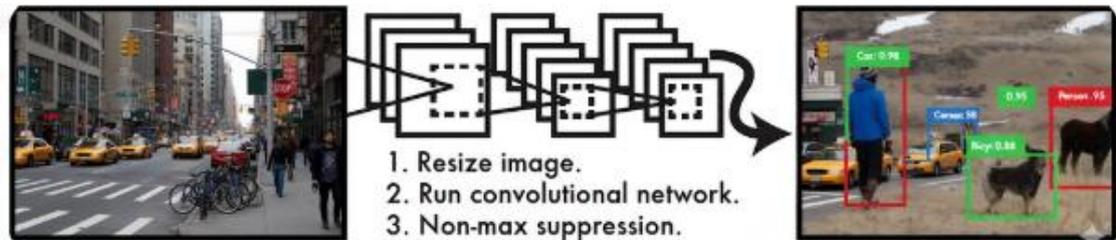


Рис. 2.5 – Принцип роботи алгоритму YOLO (You Only Look Once) для виявлення та класифікації об'єктів у реальному часі

Ключова концепція функціонування YOLO полягає у прямому прогнозуванні областей та класів об'єктів за один прохід через нейронну мережу. Алгоритм бере вхідне зображення та аналізує його, використовуючи спеціалізовані фільтри (які проходять крізь згорткову нейронну мережу), яка була попередньо навчена для ефективного виявлення цільових об'єктів (таких як транспортні засоби, пішоходи, плоди тощо).

Процес детектування включає декілька скоординованих етапів:

Згорткова нейронна мережа послідовно обробляє візуальні дані та визначає рівень імовірності присутності об'єкта у певній зоні.

У разі позитивного результату, система одночасно встановлює точні координати його обмежувального прямокутника.

Усі ці операції виконуються в межах єдиного конвеєра без потреби у повторному аналізі або багаторазовій обробці зображення. Саме ця однопрохідна архітектура пояснює назву алгоритму: You Only Look Once (Ви дивитесь лише один раз).

Архітектура нейронної мережі YOLO, що демонструє взаємозв'язки між шарами від початкового входу до кінцевого виходу, представлена на (рис. 2.6). Мережа, призначена для детектування, побудована з 24 згорткових шарів, за якими

слідують два шари повної зв'язності. Почергове застосування шарів згортки сприяє ефективному пониженню розмірності простору ознак, виділених із попередніх рівнів. Початкове навчання згорткових шарів традиційно проводиться на завданнях класифікації на наборі даних ImageNet з половинною роздільною здатністю (вхід 224×224), а потім роздільна здатність подвоюється для виконання ключового завдання — виявлення об'єктів [26].

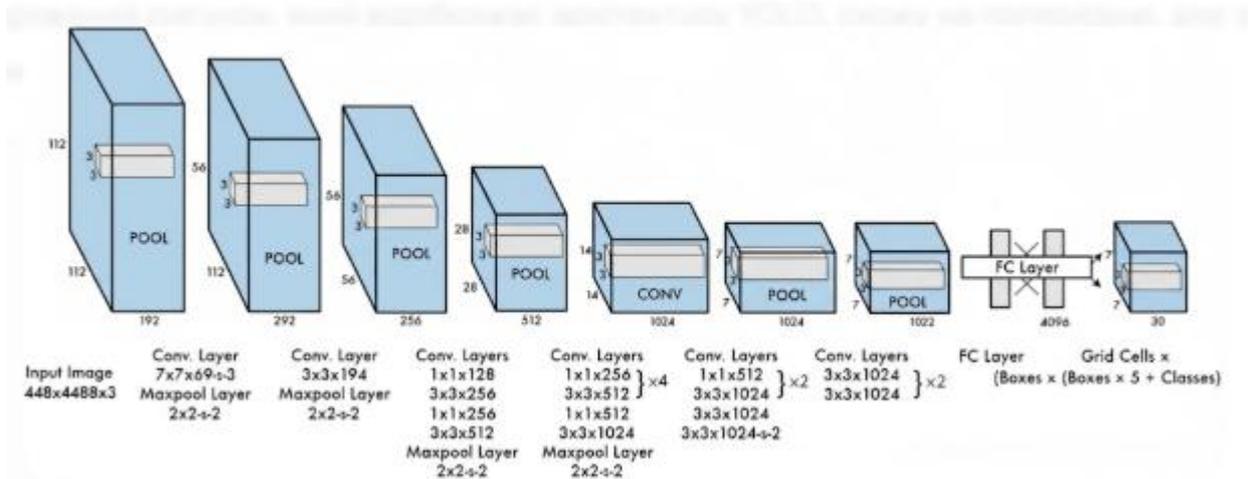


Рис. 2.6 – Архітектура згорткової нейронної мережі YOLO (You Only Look Once), що використовується для виявлення об'єктів

Сімейство моделей YOLO (You Only Look Once) систематично еволюціонувало, постійно демонструючи зростання оперативності та точності, що спричинило значний вплив на ефективність обробки візуальної інформації у режимі реального часу. З моменту публікації першої ітерації у 2015 році, алгоритм отримав численні оновлення, кожна з яких інтегрувала ключові архітектурні вдосконалення для максимізації продуктивності.

YOLOv1 стала піонерською версією, яка вперше об'єднала весь процес виявлення об'єктів у єдину нейронну мережу. Її принцип роботи базувався на розподілі зображення на сіткові комірки, одночасно прогнозуючи обмежувальні рамки (bounding boxes) та класи виявлених об'єктів. Хоча YOLOv1 виділялася високою швидкістю, її ключовим недоліком була обмежена прецизійність при детектуванні об'єктів малих розмірів.

YOLOv2, також відома під назвою YOLO9000, представила кілька фундаментальних інновацій. Ключовим моментом стало застосування мультимасштабних вхідних даних, що надало моделі можливість ефективно обробляти зображення з різною роздільною здатністю. Якщо оригінальна YOLO оперувала фіксованим розміром входу, то YOLOv2 досягла конкурентоспроможності з такими моделями, як Faster R-CNN, завдяки адаптивності до масштабів об'єктів. Це значно підвищило якість детекції та дозволило ефективно оптимізувати обмежувальні рамки через застосування згорткової нейронної мережі.

YOLOv3 продовжила шлях архітектурних удосконалень. Ця версія реалізувала кластеризацію за методом k-середніх для оптимізації початкових розмірів так званих «якірних прямокутників» (anchor boxes). Крім того, інтеграція ознакових карт (feature fusion) на різних рівнях мережі дозволила суттєво покращити роботу з дрібними об'єктами. Було оптимізовано параметри згорткових та пулінг-шарів з метою одночасного підвищення точності та зменшення загального розміру моделі.

YOLOv4 внесла інновації завдяки численним модифікаціям, включаючи впровадження функції активації Mish, агрегації ознак на основі архітектури PANet, а також низку оптимізаційних рішень для ефективного тренування глибоких нейронних мереж. Ця версія забезпечила оптимальний баланс між точністю та швидкістю, що є критично важливим для практичного використання у реальному часі.

YOLOv5 відзначилася як важливий неавторизований крок розвитку, що зосередився на швидкості навчання та гнучкості інтерфейсу для різноманітних умов застосування. Завдяки меншій кількості параметрів, ця модель забезпечила адаптивність до мобільних пристроїв та графічних процесорів (GPU).

Наступні версії, YOLOv6 та YOLOv7, були сфокусовані на подальше підвищення ефективності та продуктивності, пропонуючи кращі оптимізації для функціонування в умовах обмежених обчислювальних ресурсів. YOLOv7 на

момент свого виходу вважалася однією з найшвидших та найточніших моделей, що обійшла конкурентів у більшості тестів детекції.

YOLOv8 принесла значне оновлення у прецизійності через впровадження нових архітектурних модулів, зокрема, удосконалених механізмів уваги до ознак та гнучкої адаптації до різних обчислювальних платформ. Ця версія продемонструвала відмінні показники детекції, класифікації та сегментації у реальному часі.

Еволюція тривала з YOLOv9 та YOLOv10, які продовжили шлях оптимізації за рахунок нових механізмів уваги (attention mechanisms), що забезпечують ще більшу продуктивність при роботі з масштабними наборами даних, такими як MS COCO. Основний акцент був зроблений на поліпшенні здатності обробляти складні сценарії, включаючи низьку освітленість та високу щільність об'єктів.

На сьогоднішній день YOLOv11 є однією з найсучасніших версій, яка демонструє видатні результати на наборі даних MS COCO для виявлення об'єктів. У порівнянні з попередніми версіями, YOLOv11 поєднує поліпшену деталізацію та адаптивність до широкого спектра операційних умов. Модель забезпечує state-of-the-art показники продуктивності.

Порівняльний аналіз, представлений на рисунках 2.8 та 2.9, чітко показує, що найновіші версії YOLO (зокрема, YOLOv11) працюють значно оперативніше за більшість альтернативних методів детекції, одночасно гарантуючи високу ефективність навіть у найбільш вимогливих сценаріях [27].

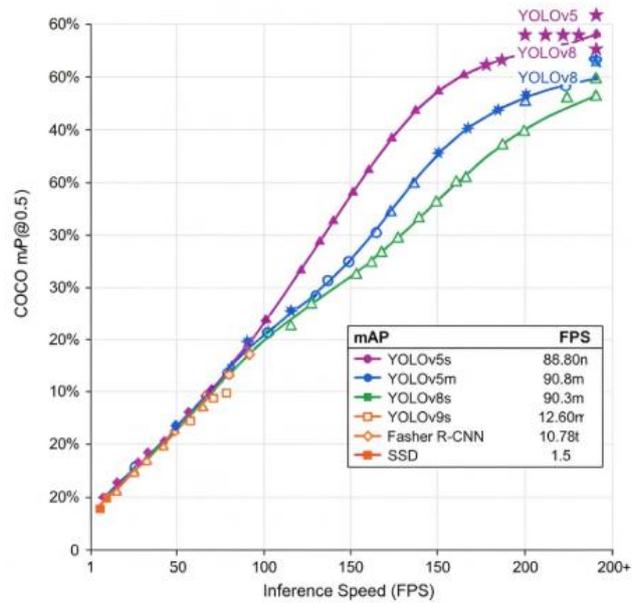


Рис. 2.8 – Порівняння продуктивності моделей сімейства YOLO з іншими алгоритмами виявлення об'єктів (співвідношення точності AP до швидкості FPS)

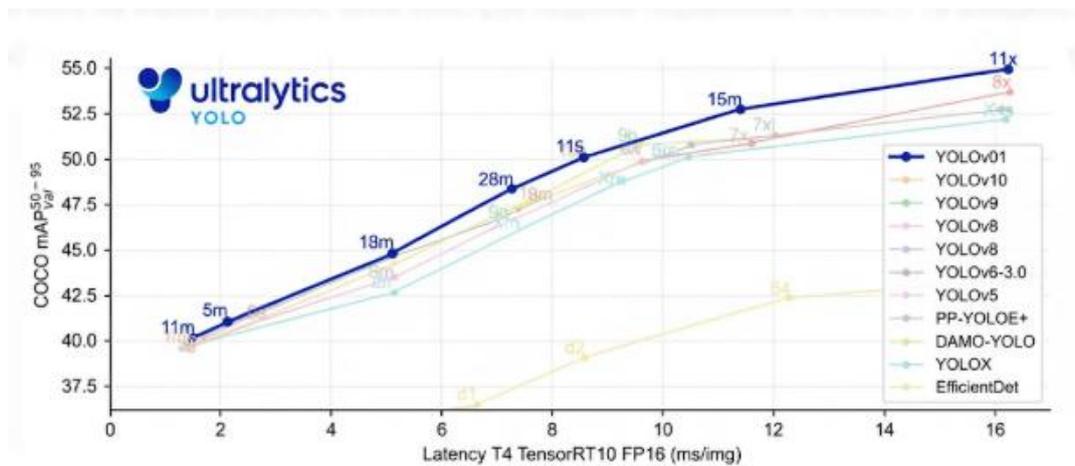


Рис. 2.9 – Співвідношення між точністю (AP) та затримкою (Latency) для останніх версій алгоритмів YOLO

2.4 Методологія підготовки та реалізації процесу детектування й класифікації об'єктів

Ефективне застосування алгоритму вимагає попереднього здійснення тренування моделі. Для успішної імплементації моделі YOLO на базі існуючого датасету необхідно послідовно виконати низку критично важливих кроків, починаючи з первинної підготовки даних та закінчуючи точним налаштуванням

гіперпараметрів тренування. Спочатку проводиться ініціалізація робочого оточення. Це включає встановлення необхідного програмного забезпечення: інтерпретатора Python та ключових бібліотек, таких як Ultralytics YOLO, PyTorch, OpenCV, NumPy та Matplotlib. Сам інструментарій YOLO (наприклад, YOLOv8) завантажується за допомогою фреймворку Ultralytics, який надає зручний інтерфейс для роботи із сучасними моделями глибокого навчання [28-30]. Наступний етап — підготовка навчального датасету. Завантажується набір даних, що містить візуальні дані (зображення) та відповідні маркування (анотації) у стандартизованому форматі (COCO або YOLO), наприклад, із платформи Kaggle. Маркування даних має бути чітким: для кожного зображення необхідно вказати координати обмежувальних прямокутників (X_{min} , Y_{min} , X_{max} , Y_{max}) та клас відповідного об'єкта. Для підвищення узагальнюючої здатності моделі та її стійкості до різних умов зйомки застосовується техніка розширення даних (аугментація), що включає масштабування, обертання та зміну яскравості.

Після валідації даних створюється файл конфігурації тренування. У ньому визначаються ключові гіперпараметри: Набір даних (Датасет): Шлях до директорій, які містять навчальні, валідаційні та тестові вибірки. Класифікація об'єктів: Чіткий перелік класів, які модель повинна розпізнавати (наприклад, Легковий автомобіль, Вантажівка, Автобус тощо). Параметри тренування: Задається кількість епох (зазвичай 10–100), розмір батчу, поріг впевненості та початкове значення швидкості навчання (learning rate). Вибір архітектури: Визначається конкретна попередня модель YOLO (наприклад, yolov8m.pt або yolov8l.pt). Після налаштування конфігурації запускається процес навчання. Навчання моделі ініціюється командою, що має наступний синтаксис:

```
yolo train model = yolov8m.pt data = dataset.yaml epochs = 50 batch = 16 imgsz = 640
```

Ця команда вказує системі YOLO використовувати попередньо навчену архітектуру та мінімізувати функцію втрат на заданому датасеті протягом визначеної кількості епох. Під час тренування модель поступово оптимізує внутрішні ваги, знижуючи рівень втрат і підвищуючи показники точності детекції. Критично важливим є моніторинг процесу навчання. Протягом кожної епохи

здійснюється обчислення різних компонентів функції втрат, таких як `box_loss` (втрати, пов'язані з точністю координат рамки), `cls_loss` (втрати на класифікації) та `dfl_loss` (втрати, що стосуються позиції рамки). Додатково, на валідаційних даних оцінюються ключові метрики: Precision (точність), Recall (повнота) та mAP (mean Average Precision). Візуалізація прогресу у формі графіків дозволяє контролювати стабільність процесу та запобігати перенавчанню [31]. Після завершення навчання навчена модель автоматично зберігається разом із повним звітом про результати, який включає графіки втрат, матрицю помилок, криві Precision-Recall та інші метрики для оцінки її продуктивності. Процес виявлення і класифікації транспортних засобів за допомогою YOLO в практичній системі поділяється на такі основні фази: Фаза детектування: Алгоритм YOLO використовується для ідентифікації транспортних засобів у кожному кадрі відеопотоку. Модель встановлює обмежувальні рамки для кожного знайденого об'єкта та присвоює йому клас (наприклад, легковий автомобіль, вантажівка тощо). Фаза класифікації (Трекінг): Для кожного виявленого транспортного засобу створюється індивідуальний трекінговий ідентифікатор. Це дозволяє відстежувати його переміщення між послідовними кадрами, що є ключовим для запобігання подвійному підрахунку та підтримки точності класифікації. Використання YOLO для аналізу транспортних засобів має значні переваги: висока точність завдяки попередньому навчанню, оперативність обробки кадрів у реальному часі та гнучкість застосування до широкого спектра задач. Ці фактори визначили YOLO як один із провідних інструментів у сфері обробки візуальної інформації. Важливо зазначити, що трекінг об'єктів у відеопотоці залишається складним завданням, особливо в умовах інтенсивного трафіку, коли відбувається перекриття (оклюзія) транспортних засобів.

2.5 Оцінка та аналіз ефективності класифікаційних алгоритмів

Для проведення дослідження методів класифікації необхідно було здійснити оцінку продуктивності розроблених моделей у контексті виявлення та

розпізнавання транспортних засобів на візуальних даних. Аналіз охоплював як кількісні, так і якісні показники ефективності алгоритмів, зокрема: точність розпізнавання, повнота, показник середньої точності (mAP - mean Average Precision), а також здатність до ідентифікації об'єктів у різноманітних умовах зйомки [32, 33]. Основними критеріями якості функціонування моделей є Precision (Точність) та Recall (Повнота). Ці метрики демонструють, наскільки коректно модель ідентифікує об'єкти (Точність) і наскільки повно вона здатна виявляти їх на зображенні (Повнота). Точність (Precision) визначається як частка тих позитивно класифікованих об'єктів, які насправді є коректними, відносно загальної кількості об'єктів, що були передбачені моделлю як позитивні:

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

де:

TP — кількість правильно передбачених позитивних об'єктів (True Positives),

FP — кількість хибно передбачених позитивних об'єктів (False Positives).

Recall (Повнота) показує, яку частку всіх реальних позитивних об'єктів модель правильно виявила:

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

де:

TP — кількість правильно передбачених позитивних об'єктів, FN — кількість об'єктів, які модель не виявила (False Negatives).

Метрика F1-Score є гармонійним середнім між Precision та Recall. Ця метрика допомагає знайти баланс між точністю та повнотою:

$$F1 - Score = \frac{Precision * Recall}{Precision + Recall} \quad (2.3)$$

Середня точність (Mean Average Precision, mAP) використовується для оцінки якості детекції об'єктів на різних порогах Intersection over Union (IoU):

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.4)$$

де:

AP_i — Average Precision для кожного класу i , N — загальна кількість класів. Значення mAP (mean Average Precision), обчислене для різних порогів IoU, слугує загальним показником точності виявлення для всіх класів об'єктів.

Перетин Intersection over Union (IoU) є метрикою, що вимірює ступінь перетину між передбаченою рамкою та реальною рамкою (ground truth).

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (2.5)$$

де:

Area of Overlap — площа спільного перетину передбаченої та справжньої рамки,

Area of Union — площа об'єднання передбаченої та справжньої рамки.

IoU зазвичай використовується для оцінки правильності детекції на заданому порозі (наприклад, $\text{IoU} > 0.5$).

Для навчання моделей класифікації та детекції використовуються функції втрат (Loss Functions):

$$L_{\text{box}} = \text{SmoothL1}(x_{\text{pred}} - x_{\text{true}}) \quad (2.6)$$

де:

x_{pred} та x_{true} — координати передбачених та справжніх рамок.

Для визначення втрат на класах використовується Cross-Entropy Loss:

$$L_{\text{cls}} = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (2.7)$$

C — кількість класів,

y_i — справжнє значення,

(\hat{y}_i) — прогнозована ймовірність належності до класу i .

Представлені формули відіграють ключову роль у всебічній оцінці якості методів класифікації та детекції об'єктів у рамках дослідження. Метрики Precision, Recall та F1-Score дозволяють оцінити продуктивність моделей на рівні окремих

класів, тоді як mAP та IoU використовуються для визначення загальної ефективності виявлення об'єктів.

Дослідження підтвердило, що точність класифікації безпосередньо залежить від кількості тренувальних прикладів для кожного класу в наборі даних. Класи, які мають більшу представленість у навчальній вибірці, закономірно демонструють вищі значення Precision та Recall. Наприклад, об'єкти, що є частими у кадрах, мають кращі показники детекції, тоді як рідкісні або мініатюрні об'єкти частіше ігноруються або класифікуються некоректно.

Було встановлено, що моделі найкраще справляються з класами, що мають чітко окреслені та стійкі ознаки (наприклад, легковий транспорт). Натомість, значні складнощі виникають при розпізнаванні об'єктів у стані оклюзії (коли вони перекривають один одного), або у випадках слабо виражених візуальних ознак, характерних для автобусів чи мотоциклів. Крім того, в умовах недостатнього освітлення та високого рівня візуального шуму продуктивність моделей суттєво знижується, що проявляється у зменшенні точності класифікації та зростанні кількості хибних позитивних спрацьовувань.

Аналіз залежності Precision-Recall від порога впевненості моделі показав типовий компроміс. Високі значення порога мінімізують хибні позитивні передбачення, забезпечуючи високу Точність, але водночас можуть призводити до пропуску об'єктів, знижуючи Повноту. Низький поріг забезпечує високий рівень Повноти, але зменшує Точність через збільшення хибних спрацьовувань. Також було виявлено, що розмір обмежувальних прямокутників та їх позиція у кадрі впливають на якість класифікації. Об'єкти малих розмірів або ті, що розташовані по краях зображення, частіше класифікуються з помилками через недостатню вираженість або спотворення їхніх ознак.

Узагальнюючи результати, можна стверджувати, що методи класифікації транспортних засобів ефективно виконують завдання виявлення та розпізнавання об'єктів у реальних сценаріях. Продуктивність моделей тісно пов'язана з якістю навчальних даних, умовами зйомки, розмірами об'єктів та візуальною складністю сцени. Для досягнення подальшого покращення результатів необхідно

зосередитися на балансуванні класів у датасеті, застосуванні аугментації даних та оптимізації гіперпараметрів навчання моделі.

Висновки до розділу 2

Систематизовано викладено теоретичні основи та архітектурні підходи до виявлення та розпізнавання транспортних засобів у сучасних системах комп'ютерного зору, акцентуючи увагу на завданнях, що вимагають оперативної обробки у режимі реального часу.

Здійснено глибоку ревізію методів детекції та класифікації об'єктів. Зокрема, детально проаналізовано еволюцію моделей серії YOLO (You Only Look Once), починаючи від YOLOv1 до YOLOv11, з висвітленням ключових архітектурних вдосконалень (таких як мультимасштабний вхід, інтеграція ознакових карт та механізми уваги), які забезпечили їхню високу продуктивність.

Виконано порівняльний аналіз продуктивності моделей серії YOLO з іншими актуальними підходами до детекції об'єктів, включаючи двоступеневі моделі (Faster R-CNN) та одноступеневі моделі (SSD), підтверджуючи перевагу YOLO у співвідношенні швидкості (FPS) та точності (mAP).

Розглянуто та детально описано ключові метрики оцінки ефективності алгоритмів детекції та класифікації, включаючи Precision, Recall, F1-Score, mAP та IoU. Додатково проаналізовано критичний вплив якості навчальних даних, порогів впевненості та балансу класів на кінцеву продуктивність моделі та її стійкість до зовнішніх факторів (як-от освітлення та оклюзія).

3 АРХІТЕКТУРА ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Деталізація програмно-технологічного стеку та інструментарію реалізації

Імплементація запропонованого методу була розподілена на ряд послідовних технологічних етапів, кожен з яких вимагав застосування специфічного набору інструментів та програмних бібліотек. Це забезпечило створення цілісної інтегрованої системи для автоматизованого моніторингу дорожнього руху.

Тренування та фіналізація моделі. На початковій фазі відбувається навчання моделі YOLO на спеціалізованому датасеті, який містить анотовані зображення транспортних засобів різних класів. Під час цього процесу модель оптимізує внутрішні вагові коефіцієнти для досягнення максимальної прецизійності виявлення та класифікації. Після успішного завершення тренування навчена модель зберігається для інтеграції у фінальну систему.

Попередня обробка відеоданих. Зчитування вхідного відеопотоку здійснюється за допомогою бібліотеки OpenCV. Кожен кадр підлягає трансформації з формату BGR у формат RGB, що є необхідною умовою для коректної роботи моделей YOLO. Далі кадр масштабується до уніфікованого розміру (наприклад, 640×640 пікселів), що підтримує оптимальний баланс між швидкістю обробки та точністю детекції.

Виявлення транспортних одиниць. Використовується попередньо навчена модель YOLO. Вхідні кадри надходять у мережу, яка формує результати, що включають обмежувальні прямокутники (bounding boxes) для кожного ідентифікованого об'єкта, присвоює йому клас (наприклад, "легковий автомобіль", "вантажний транспорт") та визначає рівень достовірності (confidence score). Цей етап дозволяє фільтрувати результати з низькою ймовірністю, підвищуючи надійність системи.

Трекінг об'єктів. Для забезпечення безперервного відстеження руху транспортних засобів між послідовними кадрами використовується алгоритм

трекінгу. Кожному виявленому об'єкту призначається унікальний ідентифікатор. Алгоритм використовує просторові координати, отримані від YOLO, для прогнозування позиції об'єктів у наступних часових точках.

Класифікація та ідентифікація. Класифікація здійснюється на основі вихідних даних, згенерованих YOLO. Завдяки навчанню на великому датасеті, система здатна розпізнавати різні типи об'єктів (автомобілі, автобуси, вантажівки, мотоцикли тощо). Для візуального розрізнення кожного класу застосовується індивідуальна колірна схема.

Візуалізація результатів у реальному часі. Фінальні результати детекції та класифікації відображаються безпосередньо на відеопотоці. Об'єкти позначаються граничними рамками, центроїдами та текстовими мітками класів. Така візуалізація є критично необхідною для систем моніторингу, забезпечуючи користувача інформацією про розпізнані транспортні засоби, їхнє розташування та клас.

Програмно-апаратна конфігурація та дані. В основі роботи системи лежать передові алгоритми YOLO, зокрема, версії YOLOv8 та YOLOv11. YOLOv8 вирізняється високою швидкістю обробки, завдяки оптимізованій архітектурі, ідеально підходячи для задач реального часу, таких як моніторинг трафіку. Тоді як YOLOv11 орієнтована на досягнення максимальної точності та деталізації, дозволяючи виявляти мініатюрні об'єкти та забезпечуючи стабільну роботу в складних умовах (погана видимість, інтенсивний рух) [34].

Задача виконувалася на апаратній конфігурації, що включала процесор AMD Ryzen 7 6850, 32 ГБ оперативної пам'яті та твердотільний накопичувач об'ємом 1 ТБ, під управлінням операційної системи Linux Ubuntu 24.04 LTS. Ця конфігурація забезпечила необхідний рівень обчислювальної потужності та стабільну роботу для виконання вимогливих обчислень.

Для програмної імплементації обрано мову Python, яка пропонує гнучкість і розширений набір інструментів. Ключовою бібліотекою для взаємодії з моделями YOLO є Ultralytics, що спрощує завантаження, адаптацію та отримання результатів детекції. Для реалізації операцій глибокого навчання задіяно фреймворк PyTorch, який є основою для тренування й тестування YOLOv8 та YOLOv11. Обробка

відеопотоку та кадрів здійснювалася за допомогою бібліотеки OpenCV, яка відповідала за попередню обробку даних (масштабування, нормалізація) та візуалізацію результатів детекції [33, 34].

Дані, використані для тестування, включали зображення, отримані з відеокадру огляду об'їзної дороги м. Київ. Це забезпечило реалістичне середовище для тестування моделей, що містить різноманітні транспортні засоби (легкові автомобілі, вантажівки). Для тренування моделей використовувався анотований датасет "Vehicle Detection: 8 Classes" з платформи Kaggle, що містить зображення транспортних засобів восьми різних класів.

Результати роботи моделей реєструвалися у текстових файлах, організованих у вигляді таблиці, що містила всі основні параметри виявленого об'єкта: координати прямокутника (X_{min} , Y_{min} , X_{max} , Y_{max}), клас об'єкта та його імовірність виявлення. Для подальшого аналізу та обробки отриманих даних використовувалися бібліотеки Pandas та NumPy. Фінальна візуалізація результатів, включно з відображенням рамок та міток на зображеннях, була реалізована за допомогою бібліотеки Matplotlib [34].

3.2 Імплементация, тренування та валідація моделі з модулем уваги СВМ

Для створення та навчання моделі глибокого навчання з інтегрованим Модулем Уваги Згорткового Блоку (СВМ) використано базову архітектуру YOLOv8 для розв'язання задачі виявлення та класифікації об'єктів. Процес розробки включав модифікацію архітектури, підготовку навчальних даних та оптимізацію гіперпараметрів.

Ініціалізація та модифікація архітектури. На першому етапі здійснюється імпортування необхідних програмних бібліотек. Використовується стек, що складається з PyTorch, Ultralytics, OpenCV, NumPy та інших пакетів для роботи з глибоким навчанням, обробки зображень та аналізу отриманих результатів.

Для реалізації вдосконаленої моделі створюється клас СВAM, який додає механізми Channel Attention (канальної уваги) та Spatial Attention (просторової уваги) до нейронної мережі. Інтеграція СВAM допомагає автоматично акцентувати увагу мережі на найбільш важливих просторових областях та каналах ознак, що критично покращує якість детекції.

Модифікація базової архітектури YOLOv8 реалізується шляхом вбудовування модуля СВAM у backbone моделі, як показано у прикладі класу:

```
class ModifiedYOLO8(nn.Module):
    def __init__(self, original_model):
        super(ModifiedYOLO8, self).__init__()
        self.backbone = original_model.model[0] # Backbone YOLO8
        self.cbam = CBAM(channels=256) # Додамо СВAM на 256
каналлах
        self.head = original_model.model[1:] # Залишок моделі
YOLO8

    def forward(self, x):
        x = self.backbone(x) # Обробка ознак у Backbone
        x = self.cbam(x) # Додавання СВAM
        for module in self.head:
            x = module(x) # Подальша обробка
        return x
```

Цей підхід, що передбачає додавання СВAM до backbone частини мережі, підвищує ефективність обробки ознак на ранніх етапах їхньої екстракції.

```
class ModifiedYOLO8(nn.Module):
    def __init__(self, original_model):
        super(ModifiedYOLO8, self).__init__()
        self.backbone = original_model.model[0] # Backbone YOLO8
        self.cbam = CBAM(channels=256) # Додамо СВAM на 256
каналлах
        self.head = original_model.model[1:] # Залишок моделі
YOLO8

    def forward(self, x):
        x = self.backbone(x) # Обробка ознак у Backbone
        x = self.cbam(x) # Додавання СВAM
        for module in self.head:
            x = module(x) # Подальша обробка
        return x
```

Принцип дії модуля СВМ впроваджує двохетапний механізм уваги: Channel Attention (Канальна увага): Модуль визначає, які канали ознак є найбільш інформативними для поточного вхідного зображення. Це досягається шляхом застосування операцій глобального усереднення (Average Pooling) та глобальної максимізації (Max Pooling) на каналному рівні, що допомагає виділити найбільш релевантні ознаки об'єктів. Spatial Attention (Просторова увага): На наступному етапі СВМ перефокусується на просторовому розташуванні об'єктів. Модуль генерує карту уваги, яка акцентує області, де з високою ймовірністю знаходяться цільові об'єкти, ефективно відсіюючи менш важливі ділянки зображення. Додавання СВМ до архітектури YOLOv8 дозволяє моделі більш вибірково обробляти ключові області та ознаки, що є особливо корисним для виявлення дрібних об'єктів або об'єктів, які знаходяться у складних умовах міського руху м. Києва. Проводиться підготовка даних для навчання, що включає завантаження датасету "Vehicle Detection: 8 Classes" з платформи Kaggle або інших релевантних даних. Далі застосовується аугментація даних (масштабування, обрізання, зміна яскравості) для покращення узагальнюючої здатності моделі.

```
from ultralytics import YOLO
from torchvision import transforms

train_transform = transforms.Compose([
    transforms.Resize((640, 640)),
    transforms.ColorJitter(brightness=0.2, contrast=0.2),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
])
```

Для тренування ініціалізується модифікована архітектура, налаштовуються оптимізатор та функція втрат, після чого запускається ітеративний процес навчання:

```

# Завантаження YOLO8
base_model = YOLO("yolov8m.pt")
model = ModifiedYOLO8(base_model)

# Параметри навчання
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
criterion = nn.CrossEntropyLoss() # Функція втрат

# Цикл навчання
for epoch in range(20): # Кількість епох
    for images, targets in train_loader: # Даталоадер
        тренувальних даних
            optimizer.zero_grad()
            outputs = model(images) # Передбачення
            loss = criterion(outputs, targets) # Розрахунок втрат
            loss.backward()
            optimizer.step()
        print(f"Epoch [{epoch+1}/20], Loss: {loss.item():.4f}")

```

Після завершення процесу тренування модель підлягає валідації на тестовому наборі даних. Для кількісної оцінки якості детекції використовувалися метрики Precision, Recall, F1-Score та mAP (середня точність).

```

# Оцінка на тестовому наборі
test_results = model.predict(source="test_images/", save=True)
print("Testing completed. Results saved.")

```

Візуалізація результатів здійснювалася шляхом накладання результатів детекції (граничних рамок, міток класів та рівнів довіри) безпосередньо на тестові зображення.

```

for result in test_results:
    result.show() # Відображення результатів

```

На (рис. 3.1.) представлені графіки результатів навчання та валідації, які підтверджують стабільність процесу тренування. Спостерігалось поступове зменшення втрат (таких як `box_loss`, `cls_loss`, `dfl_loss`) на обох наборах даних, що свідчить про покращення як визначення координат обмежувальних рамок, так і точності класифікації об'єктів. Метрики точності (Precision) та повноти (Recall) демонстрували позитивну динаміку, досягаючи значень понад 0.7 на останніх епохах. Стійке підвищення значень mAP50 та mAP50-95 підтверджує значне

покращення середньої точності моделі на різних порогах IoU, що є ключовим показником високої ефективності для задач детекції об'єктів.

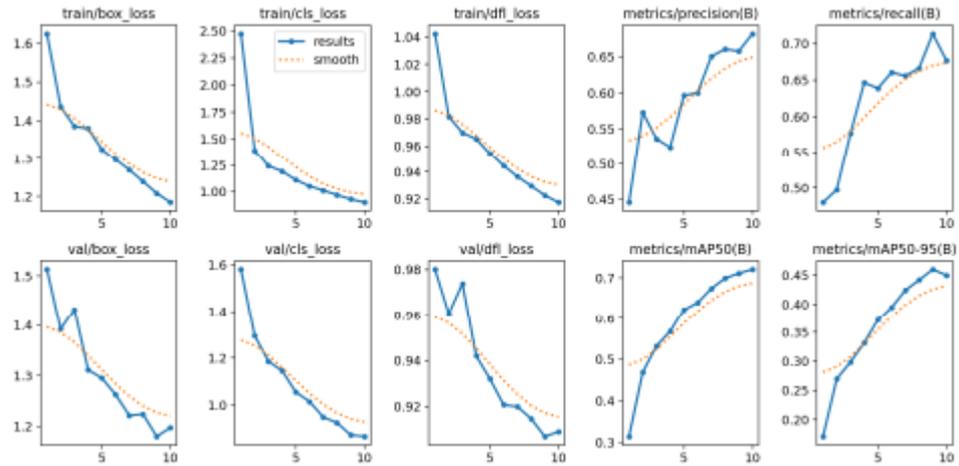


Рис. 3.1 – Зміна кривих втрат (loss) та метрик середньої точності (mAP) на валідаційному наборі даних під час навчання моделі виявлення об'єктів

На Матриці помилок (Рис. 3.2) візуалізовано розподіл коректних та хибних передбачень для кожного досліджуваного класу. Аналіз підтверджує, що клас "Car" демонструє найвищу ефективність розпізнавання з 1570 випадками істинно-позитивних класифікацій. Класи LCV (Легковий комерційний транспорт) та Multi-Axle також показують задовільні результати, незважаючи на періодичну міжкласову неточність. При цьому найбільші труднощі у розпізнаванні виникли з класом "Bus", що підтверджується найменшою кількістю правильних ідентифікацій серед усіх категорій.

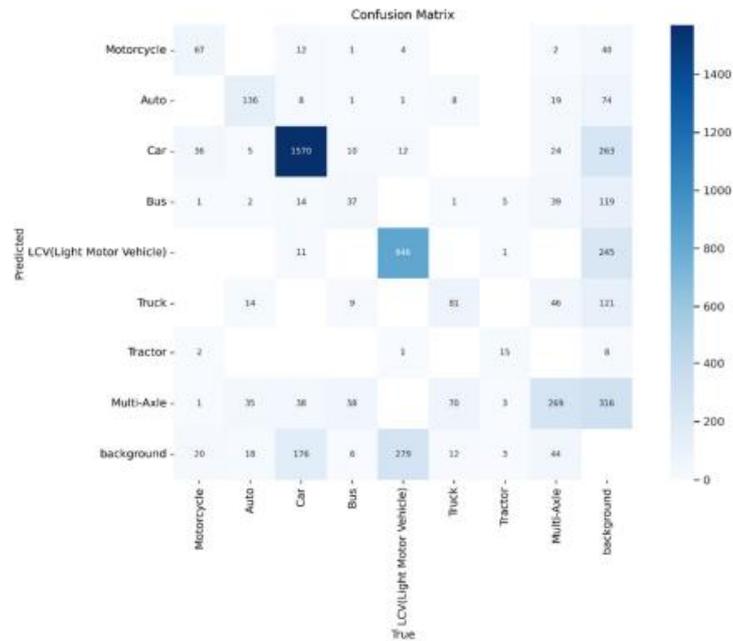


Рис. 3.2 – Матриця помилок (Confusion Matrix) для оцінки ефективності багатокласової класифікації транспортних засобів

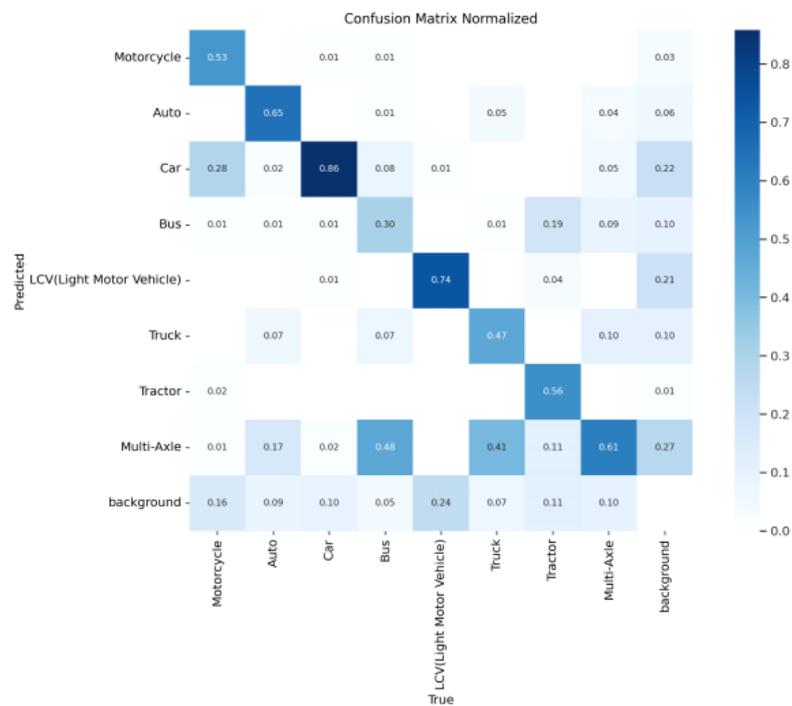


Рис. 3.3 – Нормалізована матриця помилок (Normalized Confusion Matrix) для оцінки ефективності багатокласової класифікації транспортних засобів

Аналіз нормалізованої матриці помилок (Рисунок 3.3) засвідчує високий показник Точності для класу «Легковий автомобіль» (Car), що становить 86%. Водночас, клас «Автобус» (Bus) демонструє значно нижчий рівень коректної

ідентифікації — лише 30%. Клас «Мотоцикл» (Motorcycle) досягає точності у 53%, але при цьому часто спостерігається його помилкова класифікація як «Авто» (Auto).

На графіку F1-Confidence (Рис. 3.4) візуалізовано співвідношення між метриками Точність (Precision) та Повнота (Recall) при варіативних порогах впевненості. Максимальний інтегральний показник F1-міри для всіх класів зафіксовано на рівні 0.68 при пороговому значенні 0.289. Найвища продуктивність досягнута для класів Car, Auto та LCV (Легкий комерційний транспорт), тоді як класи Bus та Motorcycle показали гірші результати, що підтверджує їхню складність для розпізнавання моделлю.

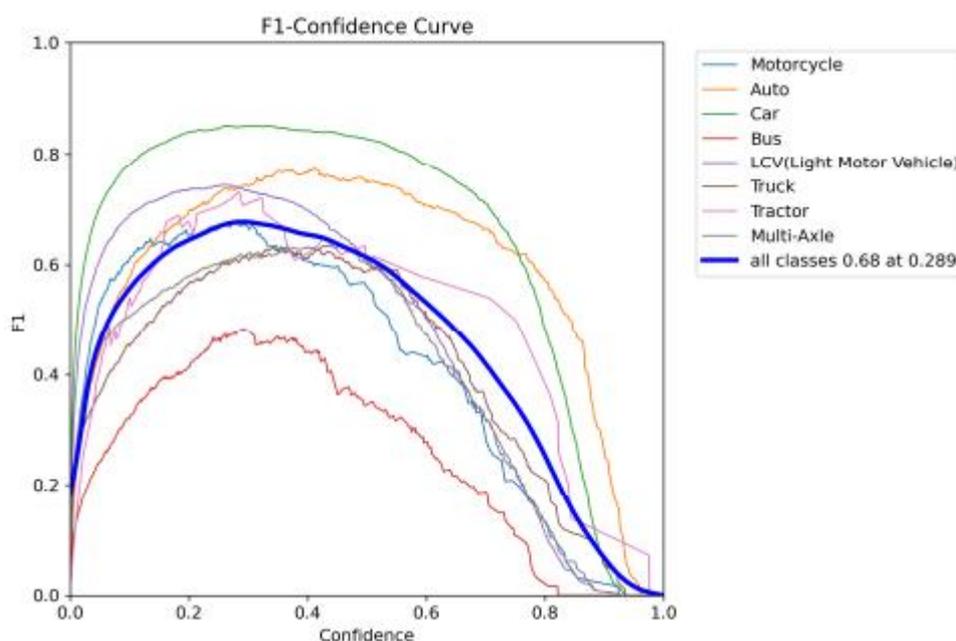


Рис. 3.4 – Графічна демонстрація балансу між Precision та Recall залежно від встановлених порогів впевненості.

(Рис. 3.5 та Рис. 3.6) відображають графіки розподілу та кореляції параметрів обмежувальних рамок: їхніх позицій, ширини та висоти. Детальний аналіз міток (розподілу об'єктів) показав, що клас Car є найбільш представленим у наборі даних, що слугує вагомим поясненням високої ефективності моделі саме для цього типу об'єктів. Спостерігається чітка концентрація обмежувальних рамок у центральній області зображень, що корелює з типовим розташуванням транспортних засобів у дорожніх сценах. Переважна більшість рамок характеризується невеликими

значеннями ширини та висоти, що підкреслює наявність викликів у сфері виявлення дрібних об'єктів.

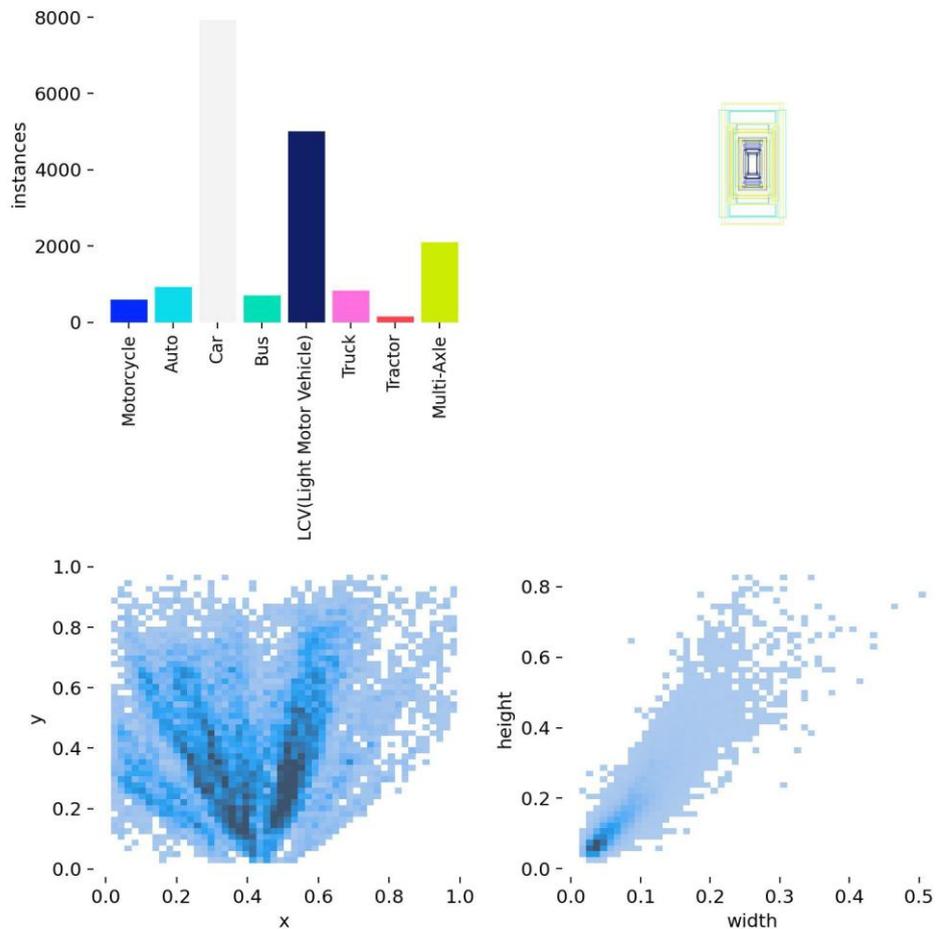


Рис. 3.5 – Розподільні характеристики позиційних координат, ширини та висоти обмежувальних рамок.

Крива Precision-Confidence (Рис. 3.7) ілюструє динаміку зміни Точності (Precision) залежно від встановленого порогу впевненості. Модель здатна досягати Precision, рівного 1.0, при високих порогах впевненості, що свідчить про надійність і достовірність ідентифікації при застосуванні жорстких критеріїв. Класи Car та LCV демонструють найвищі показники Точності, тоді як клас Bus характеризується найнижчою продуктивністю за цією метрикою.

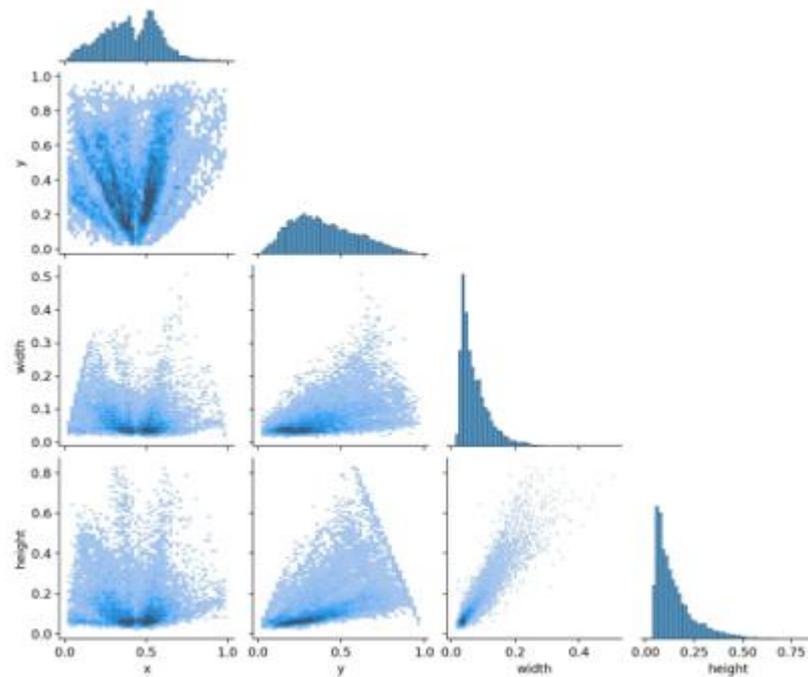


Рис. 3.6 – Графік кореляційних зв'язків між позиціями, шириною та висотою обмежувальних рамок.

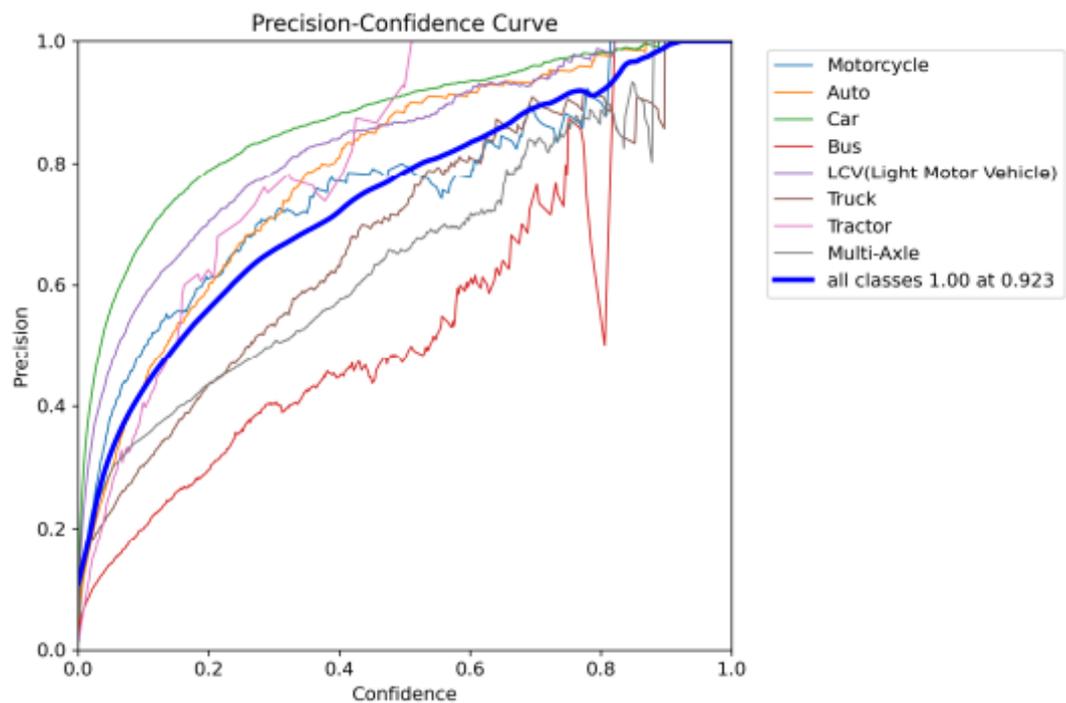


Рис. 3.7 – Крива Precision-Confidence, що відображає зміну Точності залежно від порогу впевненості, встановленого для моделі.

Графік Precision-Recall (Рис. 3.8) демонструє критичний баланс між цими двома метриками, де клас Car досягає найкращих значень (метрика становить

0.902), а Bus залишається на рівні 0.421. Узагальнене (середнє) значення становить 0.711, що розцінюється як задовільний результат для задач детекції об'єктів.

На кривій Recall-Confidence (Рис. 3.9) відображено, як змінюється Повнота (Recall) при різних порогах впевненості. Максимальне загальне значення Recall сягає 0.97, що підтверджує здатність моделі виявляти майже всі цільові об'єкти за умови встановлення низьких порогів впевненості. Найвищу Повноту демонструють класи Car та Auto, тоді як клас Bus знову показує помітно нижчий рівень результатів.

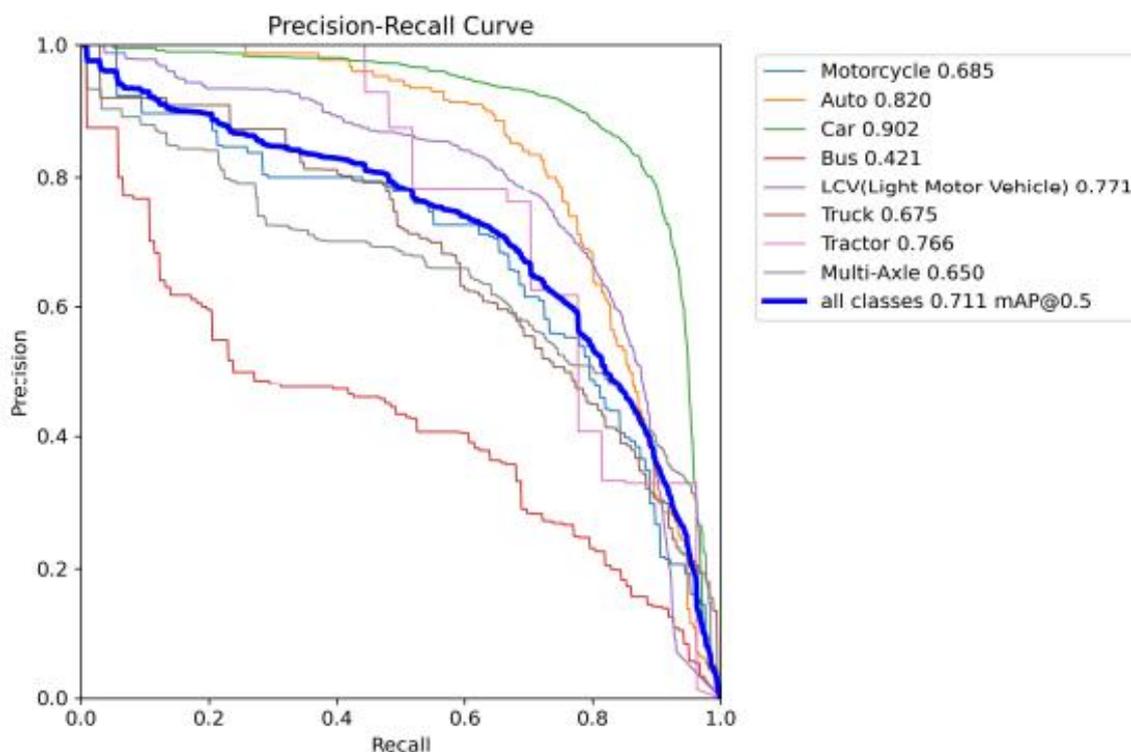


Рис. 3.8 – Крива Precision-Recall, що показує, як модель балансує між Точністю та Повнотою.

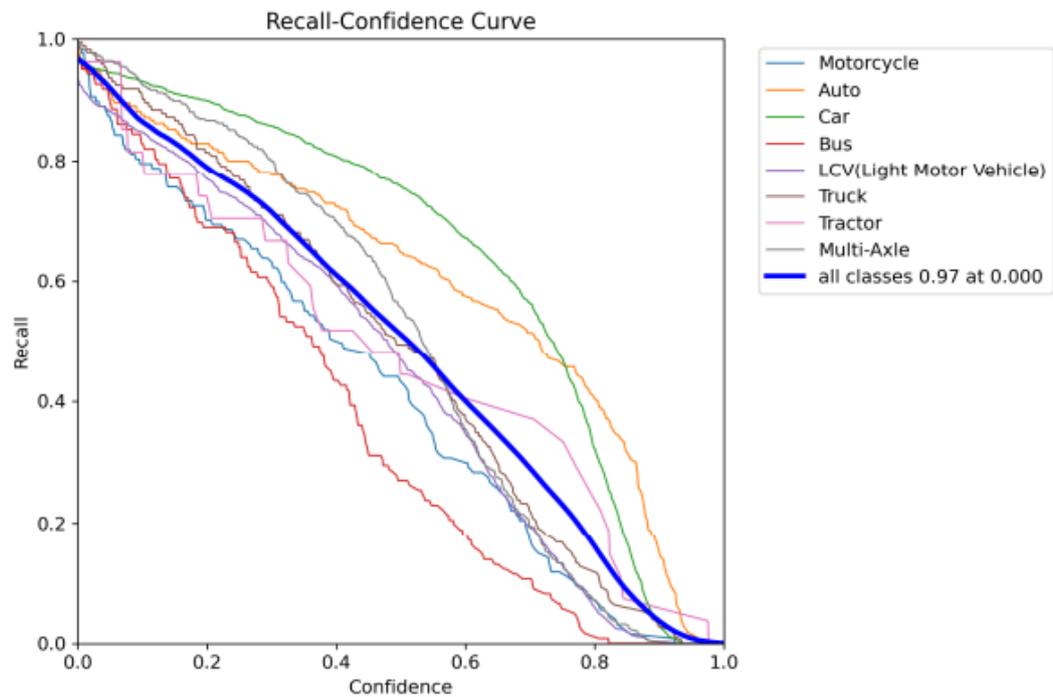


Рис. 3.9 – Крива Recall-Confidence, що відображає зміну Повноти залежно від порогу впевненості моделі.

Підсумовуючи, модель засвідчила високу продуктивність у виявленні класів Car, Auto та LCV, проте вимагає додаткового доопрацювання для класів Bus та Motorcycle, розпізнавання яких є складнішим. Загальні інтегральні метрики 0.711 та максимальний показник F1-міри = 0.68 підтверджують ефективність моделі в контексті вирішення задач детекції транспортних засобів. Для подальшого покращення результатів необхідним є збалансування навчального набору даних та проведення додаткового тонкого налаштування (тюнінгу) параметрів моделі.

3.3 Експериментальна верифікація та порівняльний аналіз продуктивності

Для проведення експериментального дослідження було використано статичний кадр (Рис. 3.10), отриманий із відеоматеріалу, опублікованого на платформі YouTube, який фіксує ділянку об'їзної дороги міста Київ. Вибір цього зображення обумовлений його репрезентативністю щодо реальних дорожніх умов та наявністю характерних елементів типової транспортної ситуації, що дозволяє

об'єктивно оцінити ефективність алгоритмів виявлення та класифікації об'єктів. Кадр містить різні категорії транспортних засобів, включаючи легкові та вантажні автомобілі, що є підставою для перевірки класифікаційної здатності алгоритмів. Висока щільність дорожнього руху та наявність частково перекритих об'єктів моделюють складні сценарії, які є типовим викликом для завдань детекції, оскільки таке перекриття ускладнює точне визначення просторових меж об'єктів.



Рис. 3.10 – Фрагмент відеокадру з огляду дорожньої інфраструктури, м. Київ, використаний для експериментальних досліджень.

Природні умови освітлення, наявність тіней та елементів інфраструктури, характерних для місця зйомки, додатково підвищують складність експерименту, забезпечуючи перевірку стійкості алгоритмів до впливу зовнішніх чинників у реальних умовах експлуатації. Таким чином, обраний кадр дозволяє всебічно перевірити можливості алгоритмів виявлення та класифікації транспортних засобів на прикладі реального дорожнього середовища, враховуючи складні умови, такі як щільний трафік, часткове перекриття об'єктів та природні умови освітлення. (Рис. 3.11) ілюструє результати виявлення та класифікації об'єктів, отримані за

допомогою базового алгоритму YOLOv8. Зіставні результати, але вже з використанням модифікованого алгоритму YOLOv8, представлені на (Рис. 3.12).

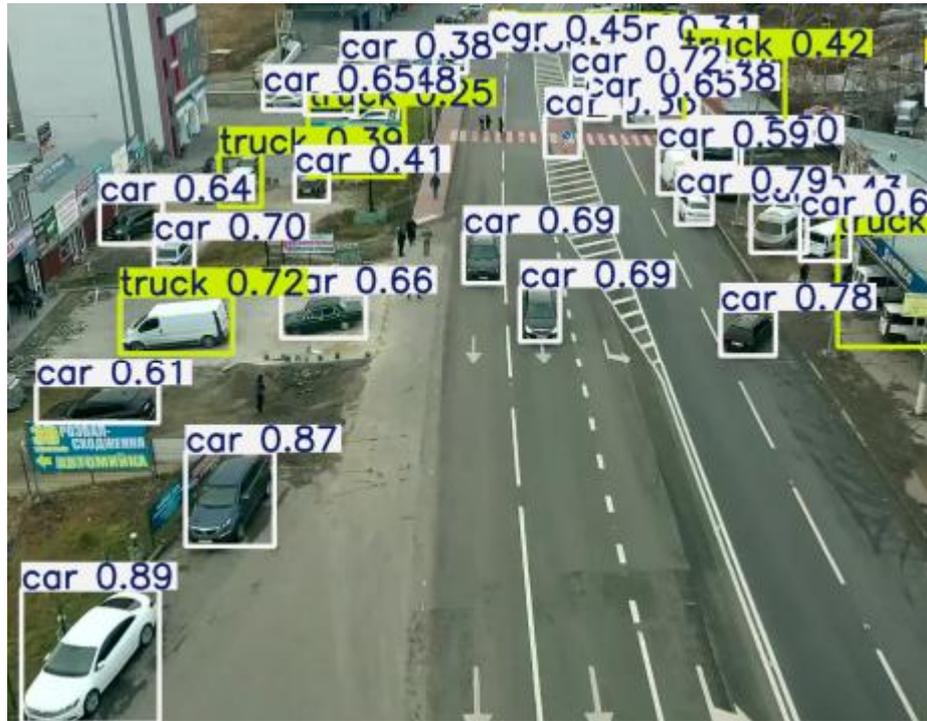


Рис. 3.11 – Виявлення і класифікація об'єктів за допомогою алгоритму YOLOv8

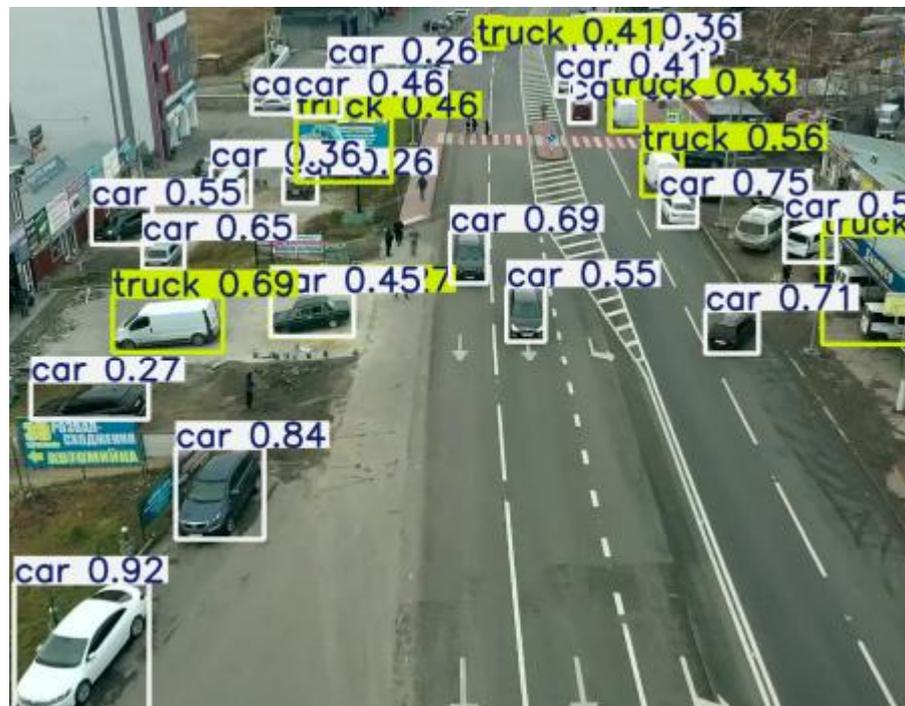


Рис. 3.12 – Виявлення і класифікація об'єктів за допомогою модифікованого алгоритму YOLOv8

Для тренування та подальшої оцінки продуктивності обох моделей, YOLOv8 та модифікованої YOLOv8, був задіяний спеціалізований набір даних "Vehicle Detection: 8 Classes", який є загальнодоступним на платформі Kaggle. Цей датасет містить попередньо анотовані зображення транспортних засобів, розділених на 8 класів, що забезпечує адекватні умови для навчання й тестування алгоритмів виявлення та класифікації об'єктів.

Відомості про всі виявлені системою об'єкти зберігаються у файлах (див. Додаток Б), які мають чітку табличну структуру: Index, Class, Probability, Xmin, Ymin, Xmax, Ymax. Кожен такий запис містить заголовок, що визначає ключові атрибути. Index є порядковим номером виділеного об'єкта, який починається з 1 і зростає для кожного нового запису. Class — це ідентифікатор класу об'єкта (наприклад, клас 2 відповідає легковому автомобілю, а клас 7 — вантажному автомобілю). Probability відображає ступінь ймовірності (впевненості), з якою алгоритм ідентифікував об'єкт, і має значення від 0 до 1, де 1 означає максимальну впевненість. Xmin та Ymin — це координати верхнього лівого кута прямокутника, що окреслює об'єкт, тоді як Xmax та Ymax визначають координати нижнього правого кута цього ж прямокутника.

```
Index, Class, Probability, Xmin, Ymin, Xmax, Ymax
```

```
1, 2, 0.89, 218.98, 876.36, 423.64, 1076.58
```

```
2, 2, 0.87, 462.36, 672.52, 594.21, 812.83
```

```
3, 2, 0.79, 1191.21, 286.72, 1245.54, 332.98
```

```
4, 2, 0.78, 1257.80, 459.34, 1339.51, 527.00
```

```
5, 2, 0.72, 1034.20, 108.74, 1065.05, 137.52
```

```
6, 7, 0.72, 364.13, 437.99, 532.00, 523.90
```

```
7, 2, 0.70, 415.04, 355.58, 475.24, 397.06
```

```
8, 2, 0.69, 875.66, 345.43, 934.15, 419.07
```

```
* * *
```

```
14, 2, 0.64, 334.77, 298.20, 428.51, 361.17
```

```
15, 2, 0.61, 240.40, 572.26, 422.22, 629.41
```

```
16, 2, 0.59, 1164.34, 218.76, 1227.63, 287.96
```

```
17, 2, 0.52, 365.29, 437.99, 532.36, 524.00
```

Для порівняльного аналізу були зіставлені результати виявлення та класифікації, отримані за допомогою стандартного та модифікованого алгоритмів YOLOv8. Базовий YOLOv8 виявив загалом 35 об'єктів, у той час як модифікована версія ідентифікувала 31 об'єкт. Ця кількісна різниця свідчить про вищу чутливість базового алгоритму, яка, втім, може потенційно призводити до хибних спрацювань. Натомість, модифікований YOLOv8 продемонстрував менше детекцій, але з вищою впевненістю для окремих випадків.

Щодо класифікації, більшість виявлень в обох моделях належить до класу 2 (легкові автомобілі). Базовий алгоритм зафіксував 29 об'єктів класу 2 і 6 об'єктів класу 7 (вантажні автомобілі). Модифікований YOLOv8 виявив 23 об'єкти класу 2 і 8 об'єктів класу 7. Перевага модифікованого YOLOv8 у кількості коректно виявлених вантажних автомобілів (клас 7) підтверджує його покращену здатність до класифікації складніших типів об'єктів.

Аналіз показників ймовірностей виявив, що максимальний показник впевненості для базового YOLOv8 становив 0.89, тоді як у модифікованому алгоритмі максимальна впевненість сягнула 0.92. Середнє значення ймовірностей для стандартного YOLOv8 склало 0.525, а для модифікованого – 0.52. Незважаючи на незначне зниження середньої ймовірності, модифікована модель продемонструвала вищу точність для об'єктів класу 7, що підтверджується більшою кількістю їх виявлень. Зіставлення координат обмежувальних рамок (код для порівняння представлений в Додатку Д) показало незначні відхилення між алгоритмами. При цьому модифікований YOLOv8 забезпечив точнішу локалізацію для вантажних автомобілів, тоді як базовий YOLOv8 частіше встановлював рамки для легкових автомобілів, включаючи об'єкти з низьким ступенем впевненості.

Зафіксовано різницю у часі виконання: обробка зображення базовим YOLOv8 зайняла 0.82 секунди, тоді як модифікований алгоритм витратив 1.04 секунди. Збільшення часу обробки для модифікованої моделі є очікуваним і пов'язане з додатковими обчисленнями, спрямованими на підвищення якості детекції складних об'єктів. Підсумовуючи, стандартний YOLOv8 показав більшу кількість виявлень за менший проміжок часу, що робить його ефективним для загальних завдань

детекції. На противагу цьому, модифікований YOLOv8 забезпечив вищу точність класифікації складніших об'єктів, як-от вантажні автомобілі, за рахунок зниження загальної кількості детекцій та збільшення часу обробки. Це позиціонує модифіковану модель як більш придатну для сценаріїв, де пріоритетом є деталізація та точність виявлення. Подальший аналіз на розширених наборах даних є необхідним для визначення впливу внесених модифікацій у різних експлуатаційних умовах, включаючи низьке освітлення та інтенсивний трафік.

Висновки до розділу 3

Проведено систематизацію та обґрунтування вибору інструментального апарату, включаючи необхідні програмні засоби, бібліотеки та фреймворки глибокого навчання, що стали основою для розробки системи виявлення та класифікації транспортних засобів на основі методів комп'ютерного зору.

З метою підвищення робастності та точності детекції об'єктів в ускладнених візуальних умовах (зокрема, при частковому перекритті, різноманітності ракурсів зйомки та мінливих умовах освітлення), було здійснено модифікацію базової архітектури YOLOv8 шляхом інтеграції механізму уваги (Attention Module).

Навчання модифікованої моделі було успішно проведено на спеціалізованому датасеті "Vehicle Detection: 8 Classes". Результати навчання продемонстрували здатність алгоритму до адаптації та ефективного функціонування в умовах, максимально наближених до реальних дорожніх сценаріїв.

Експериментальна верифікація моделей підтвердила, що внесена модифікація YOLOv8 забезпечила суттєве покращення якості виявлення та класифікації складних об'єктів (наприклад, вантажних автомобілів). Хоча час обробки зображення для модифікованої версії незначно збільшився (до 1.04 с) порівняно з базовою моделлю (0.82 с), цей компроміс виправданий досягненням вищої точності, що є критичним параметром для систем моніторингу та аналізу дорожнього руху.

Зіставлення метрик підтвердило високу продуктивність моделі для найбільш представлених у датасеті класів (Car, Auto, LCV) із загальним показником Precision-Recall (mAP) на рівні 0.711 та максимальним значенням F1-міри 0.68. При цьому виявлено необхідність подальшого доопрацювання для класів із низькою репрезентативністю або складними ознаками розпізнавання (Bus, Motorcycle), що потребує збалансування навчального набору даних.

ВИСНОВКИ

На початку роботи було здійснено ґрунтовний аналіз предметної області, що охоплює сучасні методичні підходи та алгоритми комп'ютерного зору й глибокого навчання, які застосовуються для детекції та категоризації транспортних засобів. В рамках цього аналізу проведено аналітичне дослідження існуючих методів виявлення, приділено увагу основні принципам функціонування згорткових нейронних мереж (CNN), а також виконано детальний опис інформаційно-технологічних засад та методологічних підходів. Зокрема, глибоко проаналізовано алгоритми об'єктної детекції, акцентуючи увагу на еволюції та архітектурних особливостях моделей сімейства YOLO (You Only Look Once).

На підставі цього аналізу було сформульовано основну задачу дослідження створення високопродуктивного методу виявлення та класифікації транспортних засобів. Одночасно встановлено ключові технічні вимоги до системи: висока достовірність (точність) детекції, оперативність обробки (здатність до роботи в реальному часі) та робастність (стійкість) до несприятливих зовнішніх чинників, включаючи змінне освітлення, погодні умови та просторове перекриття об'єктів.

З метою оцінки ефективності розроблених рішень проведено систематизацію та охарактеризовано основні метрики (зокрема F1-міра, mAP, Precision, Recall), що використовуються для кількісної верифікації якості процесів. Крім того, виконано порівняльний аналіз продуктивності алгоритмів YOLO з іншими провідними сучасними підходами до детекції об'єктів, такими як двохетапні Faster R-CNN та однотапні SSD. Для практичної реалізації системи на базі технологій глибокого навчання було детально описано інструментальне та програмне забезпечення, включаючи необхідні бібліотеки та середовища розробки.

В рамках виконання дослідження була модифікована архітектура нейронної мережі YOLOv8 шляхом інтеграції спеціалізованого модуля уваги (Attention Module). Це технічне рішення спрямоване на підвищення точності локалізації та ідентифікації об'єктів у візуально складних сценаріях (перекриття, різні ракурси зйомки, динамічне освітлення). Процес навчання модифікованої моделі був

успішно реалізований на спеціалізованому датасеті "Vehicle Detection: 8 Classes", що забезпечило її адаптацію до широкого діапазону реалістичних умов дорожнього середовища.

Результати експериментальної апробації підтвердили, що внесена модифікація YOLOv8 забезпечила покращену якість виявлення об'єктів. Хоча було зафіксовано незначне зростання часу виконання для модифікованої моделі, досягнене підвищення точності є критично важливим параметром для систем моніторингу дорожнього руху, що повністю компенсує збільшення часу обробки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning: Foundations and Concepts – Springer, 2023 – 649p.
2. Класифікація транспортних засобів за екологічними показниками (2023) [Електронний ресурс]. – Режим доступу: <https://ir.nmu.org.ua/handle/123456789/160680>
3. Вдосконалення методів класифікації транспортних засобів (2022) [Електронний ресурс]. – Режим доступу:- https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%B8%D1%89%D1%83%D0%BD/page1.html
4. Перспективи розвитку класифікації транспортних засобів в Україні (2021) [Електронний ресурс]. – Режим доступу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%B8%D1%89%D1%83%D0%BD/page1.html
5. Аналіз методів класифікації транспортних засобів (2020) [Електронний ресурс]. – Режим доступу: <https://mbel.kaist.ac.kr/lab/publication.html>
6. Сучасні тенденції класифікації транспортних засобів (2019) [Електронний ресурс]. – Режим доступу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%B8%D1%89%D1%83%D0%BD/page1.html
7. Транспортні системи: Класифікація, проектування, експлуатація (2022) [Електронний ресурс]. – Режим доступу: <https://nace.lursoft.lv/H/transport-i-skladirovanie>
8. Класифікація та кодування транспортних засобів (2021) [Електронний ресурс]. – Режим доступу: <https://www.kmu.gov.ua/npas/243939925>
9. Транспортне право: Класифікація транспортних засобів (2020) [Електронний ресурс]. – Режим доступу: <http://nubip.edu.ua/sites/default/files/u123/%D1%80%D0%BF%20%D1%82%D1%80%D0%B0%D0%BD%D1%81%20%D0%BF%D1%80%D0%B0%D0%B2%D0%BE.df>

10. Експлуатація та технічне обслуговування транспортних засобів: Класифікація (2019) [Електронний ресурс]. – Режим доступу: https://www.mdoffice.com.ua/ua/aMDOClassDic.html?dic_num=13
11. Огляд сучасних моделей виявлення об'єктів на основі глибокого навчання (2023) [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/2104.11892>
12. Сучасні методи сегментації зображень у комп'ютерному зорі (2022) [Електронний ресурс]. – Режим доступу: <http://arxiv.org/abs/2001.05566>
13. SIMPLE ONLINE AND REALTIME TRACKING [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/1602.00763.pdf>
14. What Is Computer Vision? [Електронний ресурс]. – Режим доступу: <https://builtin.com/machine-learning/computer-vision>
15. Computer Vision. What is Computer Vision? [Електронний ресурс]. – Режим доступу: <https://www.protex.ai/glossary/computer-vision>
16. An Introduction to Computer Vision [Електронний ресурс]. – Режим доступу: <https://www.baeldung.com/cs/computer-vision>
17. Applications of Computer Vision in AI [Електронний ресурс]. – Режим доступу: <https://www.analyticssteps.com/blogs/7-applications-computer-vision-ai>
18. Глибоке навчання для комп'ютерного зору (2021) [Електронний ресурс]. – Режим доступу: <https://www.amazon.com/Deep-Learning-Adaptive-Computation-Machine/dp/0262035618>
19. Object Recognition and How Does it Work? [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.cprime.com/resources/blog/what-is-object-recognition-and-how-does-it-work/>
20. YOLO Object Detection Explained [Електронний ресурс]. – Режим доступу: <https://www.datacamp.com/blog/yolo-object-detection-explained>
21. What Is YOLO Algorithm? [Електронний ресурс]. – Режим доступу: <https://www.baeldung.com/cs/yolo-algorithm>
22. TrafficVision [Електронний ресурс]. – Режим доступу: <http://www.coralsales.com/products/its/camerasandddetection/trafficvision/>

23. Object Tracking [Електронний ресурс]. – Режим доступу до ресурсу: <https://encord.com/glossary/object-tracking-definition/>
24. The Complete Guide to Object Tracking [Електронний ресурс]. – Режим доступу до ресурсу: <https://encord.com/blog/object-tracking-guide/#h1>
25. Introduction to the Top Object Tracking Techniques [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.tasq.ai/blog/object-tracking/>
26. Комп'ютерний зір для автономних транспортних засобів: огляд (2019) [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/1704.05519>
27. Ultralytics. Benchmark Mode — Ultralytics Documentation [Електронний ресурс]. – URL: <https://docs.ultralytics.com/modes/benchmark/>
28. OpenCV Tutorial | OpenCV using Python [Електронний ресурс]. – Режим доступу: <https://www.javatpoint.com/opencv>
29. Комп'ютерний зір: основи та алгоритми (2023) [Електронний ресурс]. – Режим доступу: <http://szeliski.org/Book/>
30. Комп'ютерний зір: сучасні методи та застосування (2022) [Електронний ресурс]. – Режим доступу: <https://www.amazon.com/Computer-Vision-Modern-Approach-2nd/dp/013608592X>
31. Azhad Zuraimi, Fadhlan Hafizhelmi Kamaru Zaman, "Vehicle Detection and Tracking using YOLO and DeepSORT", in 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) DOI:10.1109/ISCAIE51753.2021.9431784
32. M. Ravichandran, K. Laxmikant and A. Muthu, "Efficient Vehicle Detection and Classification using YOLO v8 for Real-Time Applications," 2023 Global Conference on Information Technologies and Communications (GCITC), Bangalore, India, 2023, pp. 1- 5, doi: 10.1109/GCITC60406.2023.10426587.
33. J. Karangwa, J. Liu and Z. Zeng, "Vehicle Detection for Autonomous Driving: A Review of Algorithms and Datasets," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 11, pp. 11568-11594, Nov. 2023, doi: 10.1109/TITS.2023.3292278.

34. Michael Abebe Berwo, Asad Khan, Yong Fang, Hamza Fahim, Shumaila Javaid, Jabar Mahmood, Zain Ul Abideen, Syam M.S. «Deep Learning Techniques for Vehicle Detection and Classification from Images/Videos: A Survey», in *Sensors* 2023, 23(10), 4832; <https://doi.org/10.3390/s23104832>

Додаток А

Код модифікованого методу

```
from ultralytics import YOLO
import time
import torch
import torch.nn as nn
import torch.nn.functional as F
# CBAM Module
class CBAM(nn.Module):
    def __init__(self, channels, reduction=16):
        super(CBAM, self).__init__()
        # Channel Attention
        self.channel_attention = nn.Sequential(
            nn.AdaptiveAvgPool2d(1),
            nn.Conv2d(channels, channels // reduction, 1,
                bias=False),
            nn.ReLU(),
            nn.Conv2d(channels // reduction, channels, 1,
                bias=False), nn.Sigmoid()
        )
        # Spatial Attention
        self.spatial_attention = nn.Sequential(
            nn.Conv2d(2, 1, 7, padding=3, bias=False),
            nn.Sigmoid()
        )
    def forward(self, x):
        # Channel Attention
        ca = self.channel_attention(x)
        x = x * ca
        # Spatial Attention
        sa = torch.cat([torch.mean(x, dim=1, keepdim=True),
            torch.max(x, dim=1, keepdim=True)[0]], dim=1)
        sa = self.spatial_attention(sa)
        x = x * sa
        return x
# Модифікація моделі YOLO8 з CBAM
```

```

class ModifiedYOLO8(nn.Module):
    def __init__(self, original_model):
        super(ModifiedYOLO8, self).__init__()
        self.backbone = original_model.model[0] # Backbone
        # моделі YOLO8
        self.cbam = CBAM(channels=256) # Вибір кількості
        # каналів для CBAM (адаптуйте при потребі)
        self.head = original_model.model[1:] # Залишок моделі
        # YOLO8
    def forward(self, x):
        x = self.backbone(x)
        x = self.cbam(x)
        for module in self.head:
            x = module(x)
        return x

# Завантаження оригінальної моделі YOLO8
original_model = YOLO("yolov8m.pt")
yolo_model = ModifiedYOLO8(original_model)
# Detect objects from classes
classes = [1, 2, 3, 5, 7]
# Вимірювання часу виконання
start_time = time.time()
# Виконання передбачення з модифікованою моделлю
results = original_model.predict(source="output.jpg",
classes=classes)
end_time = time.time()
execution_time = end_time - start_time
# Збереження результатів
output_file = "detectionsYolo8_CBAM.txt"
with open(output_file, "w") as f:
    f.write("Index,Class,Probability,Xmin,Ymin,Xmax,Ymax\n") #
    # Заголовок CSV
    for i, (box, cls, conf) in
    enumerate(zip(results[0].boxes.xyxy, results[0].boxes.cls,
results[0].boxes.conf)):
        xmin, ymin, xmax, ymax = box.tolist()
        class_id = int(cls)
        probability = float(conf)

```

```
f.write(f"{i+1},{class_id},{probability:.2f},{xmin:.2f},{ymin:.2f},{
xmax:.2f},{ymax:.2f}\n")
    f.write(f"\nExecution Time: {execution_time:.2f} seconds\n")
print(f"Detections saved to {output_file}")
results[0].show()
```

Додаток Б

Інформація про виділені об'єкти

Результати роботи методу YOLOv8

```
Index,Class,Probability,Xmin,Ymin,Xmax,Ymax
1,2,0.89,218.98,876.36,423.64,1076.58
2,2,0.87,462.36,672.52,594.21,812.83
3,2,0.79,1191.21,286.72,1245.54,332.98
4,2,0.78,1257.80,459.34,1339.51,527.00
5,2,0.72,1034.20,108.74,1065.05,137.52
6,7,0.72,364.13,437.99,532.00,523.90
7,2,0.70,415.04,355.58,475.24,397.06
8,2,0.69,875.66,345.43,934.15,419.07
9,2,0.69,959.22,425.20,1018.53,505.36
10,2,0.66,604.85,434.00,730.77,500.16
11,2,0.66,1375.70,319.90,1450.80,386.34
12,2,0.65,577.47,136.09,640.15,163.56
13,2,0.65,1055.65,144.17,1093.79,177.74
14,2,0.64,334.77,298.20,428.51,361.17
15,2,0.61,240.40,572.26,422.22,629.41
16,2,0.59,1164.34,218.76,1227.63,287.96
17,2,0.52,365.29,437.99,532.36,524.00
18,2,0.48,639.46,138.69,716.10,170.83
19,2,0.45,916.15,21.70,953.50,65.82
20,2,0.43,1302.01,299.14,1386.11,374.05
21,7,0.42,1203.52,84.61,1353.07,214.37
22,2,0.41,624.97,255.80,675.01,298.21
23,7,0.39,512.65,228.83,575.99,306.73
24,2,0.38,692.89,87.87,754.14,113.15
25,2,0.38,1114.92,134.45,1162.03,187.33
26,2,0.36,996.68,170.65,1048.64,233.28
27,2,0.36,1616.59,164.62,1701.37,194.58
28,2,0.35,1100.84,100.08,1136.35,146.24
29,2,0.31,1082.71,63.61,1114.66,101.63
30,7,0.30,1431.30,344.72,1579.73,513.88
31,2,0.30,1206.79,211.58,1279.39,242.60
32,2,0.30,832.32,77.57,880.60,109.50
33,2,0.26,1210.22,213.54,1288.38,247.69
34,7,0.26,916.28,19.71,953.98,65.63
35,7,0.25,644.06,157.02,787.59,262.53
Execution Time: 0.82 seconds
```

Результати роботи модифікованого методу YOLOv8

Index, Class, Probability, Xmin, Ymin, Xmax, Ymax

1, 2, 0.92, 219.00, 877.37, 422.43, 1075.35
2, 2, 0.84, 462.15, 672.84, 595.78, 811.13
3, 2, 0.81, 1865.73, 845.16, 1919.32, 987.43
4, 2, 0.75, 1189.68, 287.32, 1247.81, 335.26
5, 2, 0.71, 1259.34, 461.01, 1339.55, 526.84
6, 7, 0.69, 367.44, 440.39, 533.33, 524.54
7, 2, 0.69, 875.98, 342.70, 931.90, 420.41
8, 2, 0.65, 412.60, 355.85, 475.65, 399.02
9, 7, 0.56, 1163.31, 219.39, 1224.32, 288.18
10, 2, 0.55, 335.15, 302.30, 427.21, 361.47
11, 2, 0.55, 960.63, 425.82, 1018.31, 508.94
12, 2, 0.52, 1378.68, 321.37, 1456.28, 388.41
13, 2, 0.46, 640.37, 137.64, 712.04, 168.36
14, 7, 0.46, 643.33, 170.29, 787.03, 267.26
15, 2, 0.45, 605.44, 434.24, 730.88, 500.92
16, 2, 0.43, 1782.52, 308.96, 1858.55, 358.93
17, 2, 0.41, 1034.77, 110.38, 1063.67, 137.22
18, 7, 0.41, 915.68, 19.00, 954.38, 65.90
19, 2, 0.38, 576.54, 136.33, 638.80, 164.59
20, 2, 0.36, 515.43, 244.46, 575.80, 305.71
21, 2, 0.36, 1081.08, 52.76, 1115.74, 101.00
22, 7, 0.33, 1115.86, 136.38, 1162.48, 190.89
23, 2, 0.33, 1054.82, 142.01, 1093.77, 178.87
24, 2, 0.31, 1116.11, 136.14, 1162.27, 190.62
25, 7, 0.31, 1435.40, 346.21, 1578.06, 512.04
26, 2, 0.28, 1616.99, 166.83, 1700.09, 195.48
27, 7, 0.27, 604.28, 433.81, 730.42, 501.50
28, 2, 0.27, 244.08, 571.93, 421.71, 629.94
29, 2, 0.26, 691.44, 88.63, 753.15, 113.94
30, 2, 0.26, 624.20, 256.16, 675.15, 299.44
31, 2, 0.25, 1056.42, 35.77, 1109.98, 73.01

Execution Time: 1.04 seconds

Додаток В

Код для порівняння результатів

```
import csv
import math

def iou(box1, box2, delta=10):
    """
    Розрахунок Intersection over Union (IoU) для двох
    прямокутників із похибкою координат.
    delta - величина похибки (в пікселях) для +/- корекції
    координат.
    """
    x1 = max(box1[0] - delta, box2[0] - delta)
    y1 = max(box1[1] - delta, box2[1] - delta)
    x2 = min(box1[2] + delta, box2[2] + delta)
    y2 = min(box1[3] + delta, box2[3] + delta)

    inter_area = max(0, x2 - x1) * max(0, y2 - y1)
    box1_area = (box1[2] - box1[0]) * (box1[3] - box1[1])
    box2_area = (box2[2] - box2[0]) * (box2[3] - box2[1])

    union_area = box1_area + box2_area - inter_area
    return inter_area / union_area if union_area > 0 else 0

def read_detections(file_path):
    """Зчитування детекцій із файлу detections.txt."""
    detections = []
    with open(file_path, "r") as f:
        reader = csv.reader(f)
        next(reader) # Пропустити заголовок
        for row in reader:
            if len(row) < 7: # Пропустити рядки без даних
                continue
            index = int(row[0])
            class_id = int(row[1])
            probability = float(row[2])
            xmin, ymin, xmax, ymax = map(float, row[3:7])
            detections.append({"index": index, "class":
                class_id, "probability":
probability, "box": (xmin, ymin, xmax, ymax)})
    return detections

def compare_detections(file1, file2, iou_threshold=0.5,
prob_threshold=0.1, delta=10):
    """Порівняння детекцій між двома файлами з урахуванням
    похибки в координатах."""
```

```

detections1 = read_detections(file1)
detections2 = read_detections(file2)

matches = []
for det1 in detections1:
    for det2 in detections2:
        iou_score = iou(det1["box"], det2["box"],
                        delta)
        prob_diff = abs(det1["probability"] -
                        det2["probability"])
        if det1["class"] == det2["class"] and
iou_score > iou_threshold and prob_diff <= prob_threshold:
            matches.append({
                "file1_index": det1["index"],
                "file2_index": det2["index"],
                "class": det1["class"],
                "iou": iou_score,
                "prob_diff": prob_diff
            })

    return matches

def save_matches_to_file(matches, output_file):
    """Зберігає результати збігів у файл."""
    with open(output_file, "w", newline="") as f:
        writer = csv.writer(f)
        # Заголовок файлу
        writer.writerow(["File1_Index", "File2_Index",
"Class", "IoU", "Probability_Difference"])
        # Запис результатів
        for match in matches:
            writer.writerow([
                match["file1_index"],
                match["file2_index"],
                match["class"],
                f"{match['iou']:.2f}",
                f"{match['prob_diff']:.2f}"
            ])

"""

# Шляхи до файлів
file1 = "detectionsYolo8File0.txt"
file2 = "detectionsYolo8mFile0.txt"
output_file = "detectionsYolo8AndYolo8mFile0Out.csv"

"""

# Шляхи до файлів
file1 = "detectionsYolo8File4.txt"

```

```
file2 = "detectionsYolo8mFile4.txt"
output_file = "detectionsYolo8AndYolo8mFile4Out.csv"

# Порівняння
matches = compare_detections(file1, file2, delta=10)

# Вивід результатів
print(f"Знайдено {len(matches)} збігів:")
for match in matches:
    print(f"File1 Index: {match['file1_index']}, File2 Index:
{match['file2_index']}, "
        f"Class: {match['class']}, IoU: {match['iou']:.2f},
        Prob
        Diff: {match['prob_diff']:.2f}")

# Збереження результатів у файл
save_matches_to_file(matches, output_file)
print(f"Результати збережено у файл: {output_file}")
```

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інформаційних систем та технологій

«КОМП'ЮТЕРНИЙ ЗІР У СИСТЕМАХ АВТОМАТИЧНОЇ ІДЕНТИФІКАЦІЇ ТА КАТЕГОРИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ»

Виконав: студент групи ІСДМ-61 Марк ЖИЛІН

Науковий керівник: PhD Валентина ДАНИЛЬЧЕНКО

МЕТА РОБОТИ ТА АКТУАЛЬНІСТЬ

Мета роботи - розробка вдосконаленого методу автоматизованого виявлення та класифікації транспортних засобів.

Об'єкт дослідження - виступає процедура автоматичного ідентифікування та категоризації рухомих транспортних засобів у відеозаписах.

Предметом дослідження є - модифіковані алгоритмічні рішення та методи, що використовуються для автоматизації моніторингу дорожнього руху, спрямовані на збільшення точності та швидкості обробки даних.

Актуальність теми зумовлена об'єктивною потребою у створенні надійних та високопродуктивних систем моніторингу дорожнього руху, здатних в автоматичному режимі ідентифікувати види транспортних засобів, відстежувати їхню траєкторію та гарантувати обробку даних у режимі реального часу.

ЗАВДАННЯ ПРОЕКТУ

1. Проаналізувати наявні наукові підходи до виявлення та класифікації транспорту у відеопотоках, ідентифікувавши ключові виклики для малорозмірних та перекритих об'єктів.
2. Розробити методологію модифікації архітектури YOLOv8 шляхом вбудовування модуля для покращення детекції складних об'єктів.
3. Реалізувати запропоновану модифіковану нейронну мережу та створити на її базі функціональну систему виявлення транспорту.
4. Провести порівняльне експериментальне оцінювання продуктивності розробленої системи з класичними та стандартними моделями глибокого навчання.
5. Підтвердити ефективність системи в складних експлуатаційних умовах (низьке освітлення, інтенсивний трафік).

ПІДХОДИ ДО ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ТРАНСПОРТУ У ВІДЕОПОТОКАХ

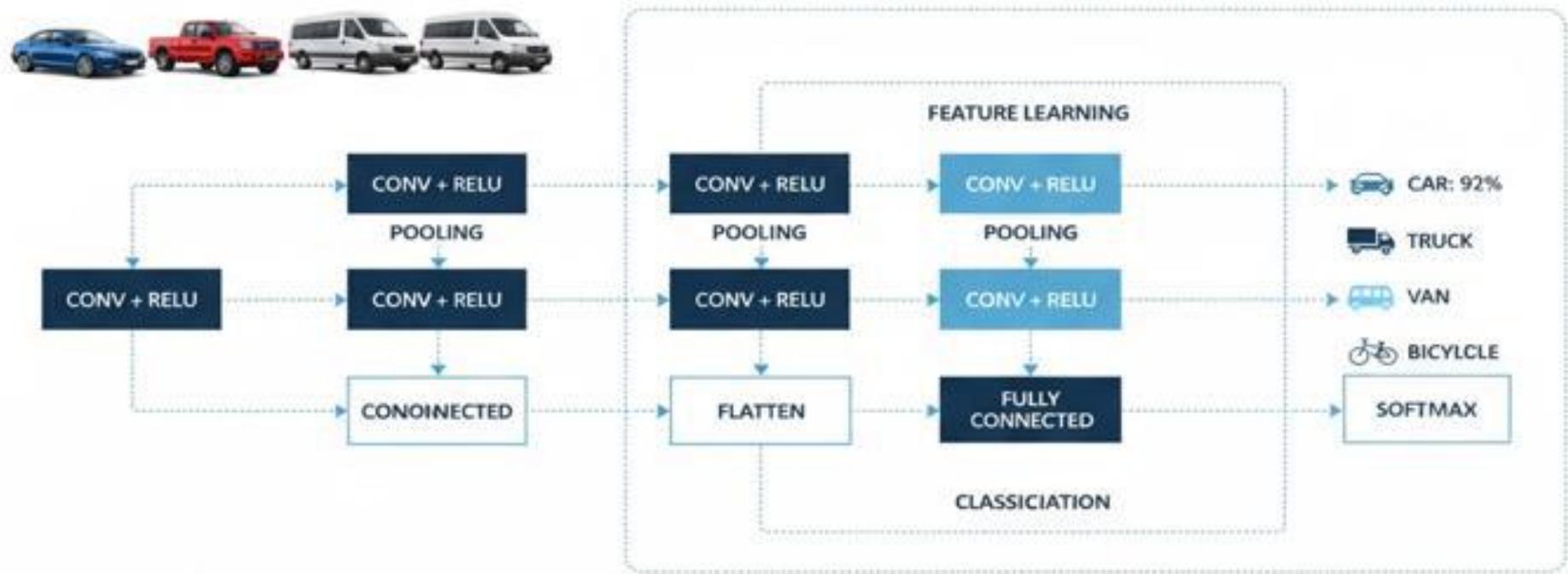


Рис. 4.1 – Загальна архітектура згорткової нейронної мережі для класифікації

ОБҐРУНТУВАННЯ МОДИФІКАЦІЇ ТА ВИБІР РІШЕННЯ

Модифікація YOLOv8

1. Проблема	Низька достовірність детекції малорозмірних та перекритих об'єктів у стандартній YOLOv8.
2. Мета Модифікації	Підвищення точності локалізації та ідентифікації об'єктів у візуально складних сценаріях (перекриття, різні ракурси, динамічне освітлення).
3. Технічне Рішення	Інтеграція спеціалізованого модуля уваги (Attention Module) в архітектуру нейронної мережі YOLOv8.

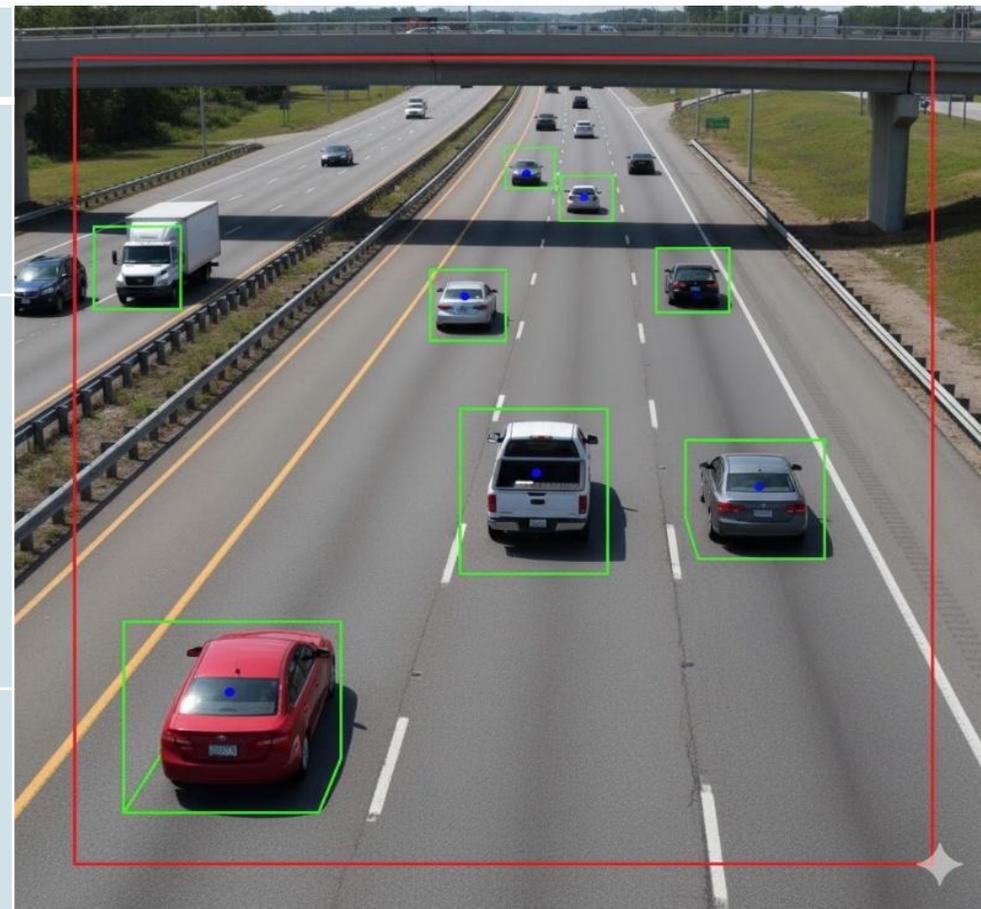


Рис. 5.1 – Обмеження стандартних методів детекції при виявленні малорозмірних та перекритих об'єктів.

СХЕМА МОДИФІКОВАНОЇ АРХІТЕКТУРИ

6

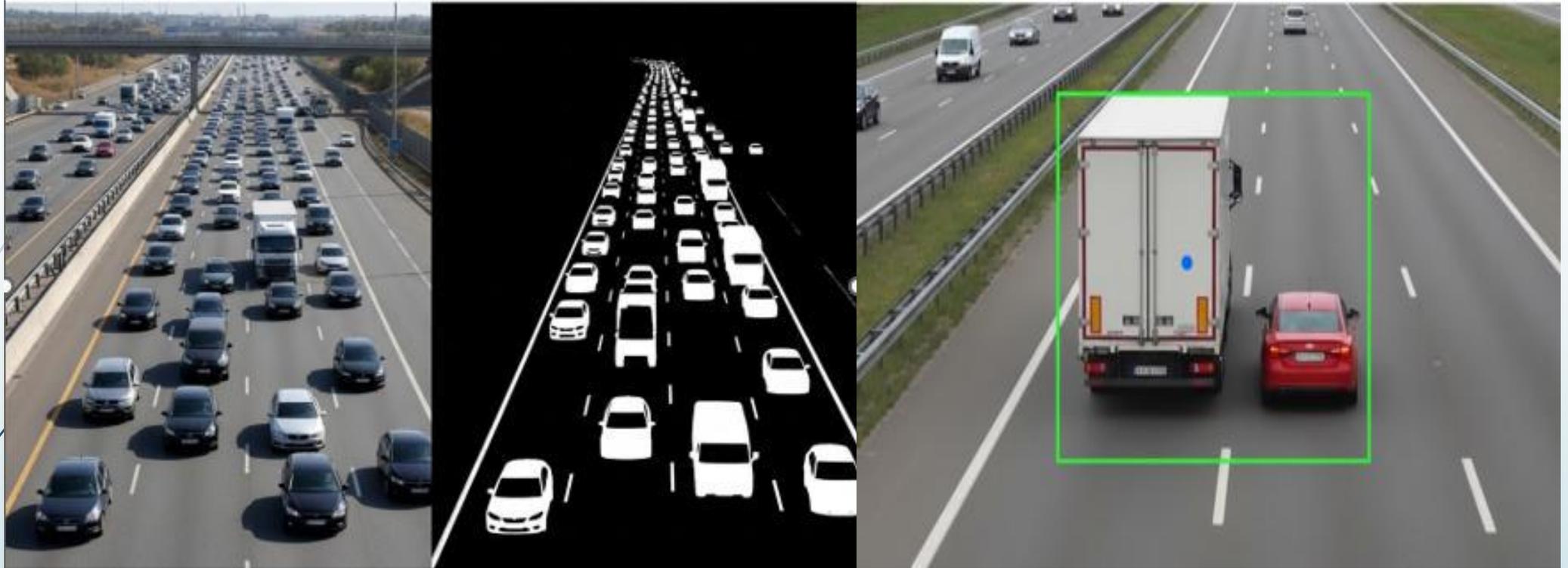


Рис. 6.1 – Результат застосування алгоритму MOG2: виділення рухомих об'єктів

Рис. 6.2 – Приклад проблемного сценарію: накладання

Методологія модифікації архітектури YOLOv8 полягала в інтеграції спеціалізованого модуля уваги (Attention Module), спрямованого на підвищення точності локалізації та ідентифікації об'єктів у складних сценаріях (перекриття, різні ракурси). Це технічне рішення забезпечило механізм для фокусування мережі на найбільш інформативних ознаках, що є ключовим для покращення детекції малорозмірних та перекритих транспортних засобів.

РЕАЛІЗАЦІЯ ТА НАВЧАННЯ МОДИФІКОВАНОЇ МОДЕЛІ

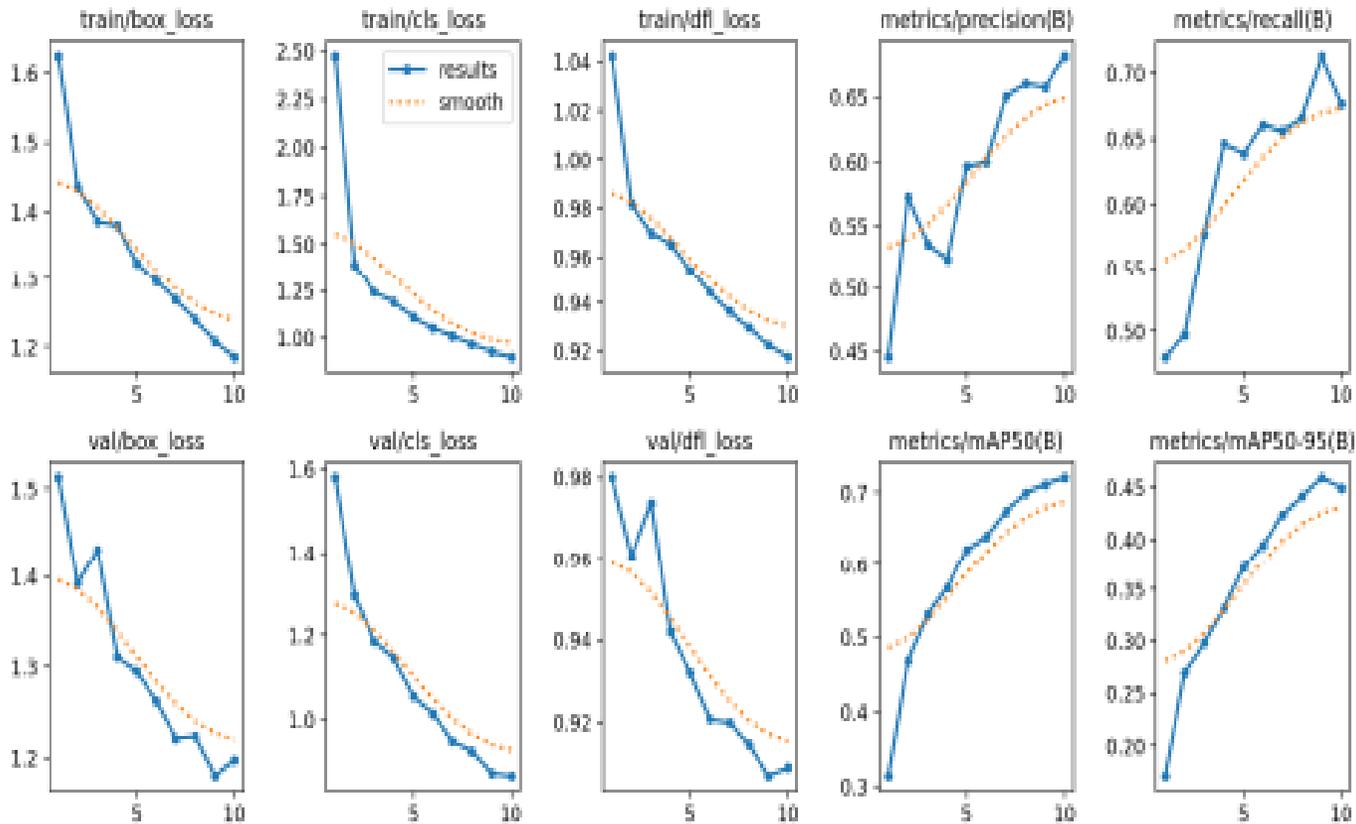


Рис. 7.1 – Графіки результатів навчання

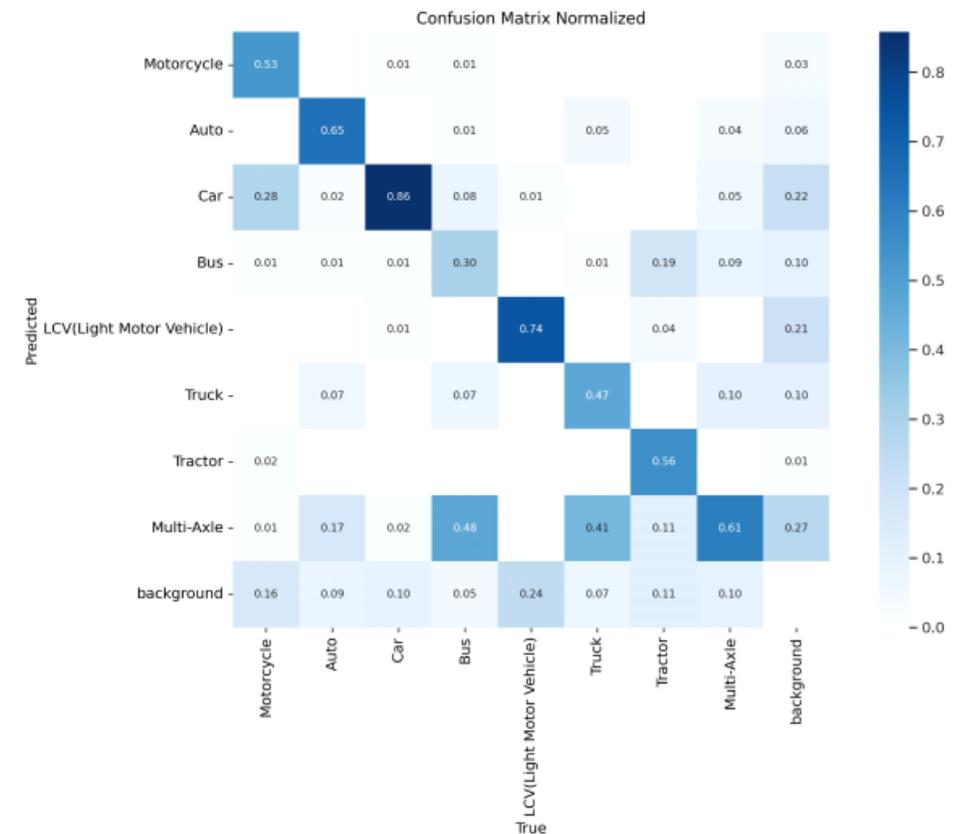


Рис. 7.2 – Матриця помилок (Нормалізована)

ПОРІВНЯЛЬНЕ ЕКСПЕРИМЕНТАЛЬНЕ ОЦІНЮВАННЯ

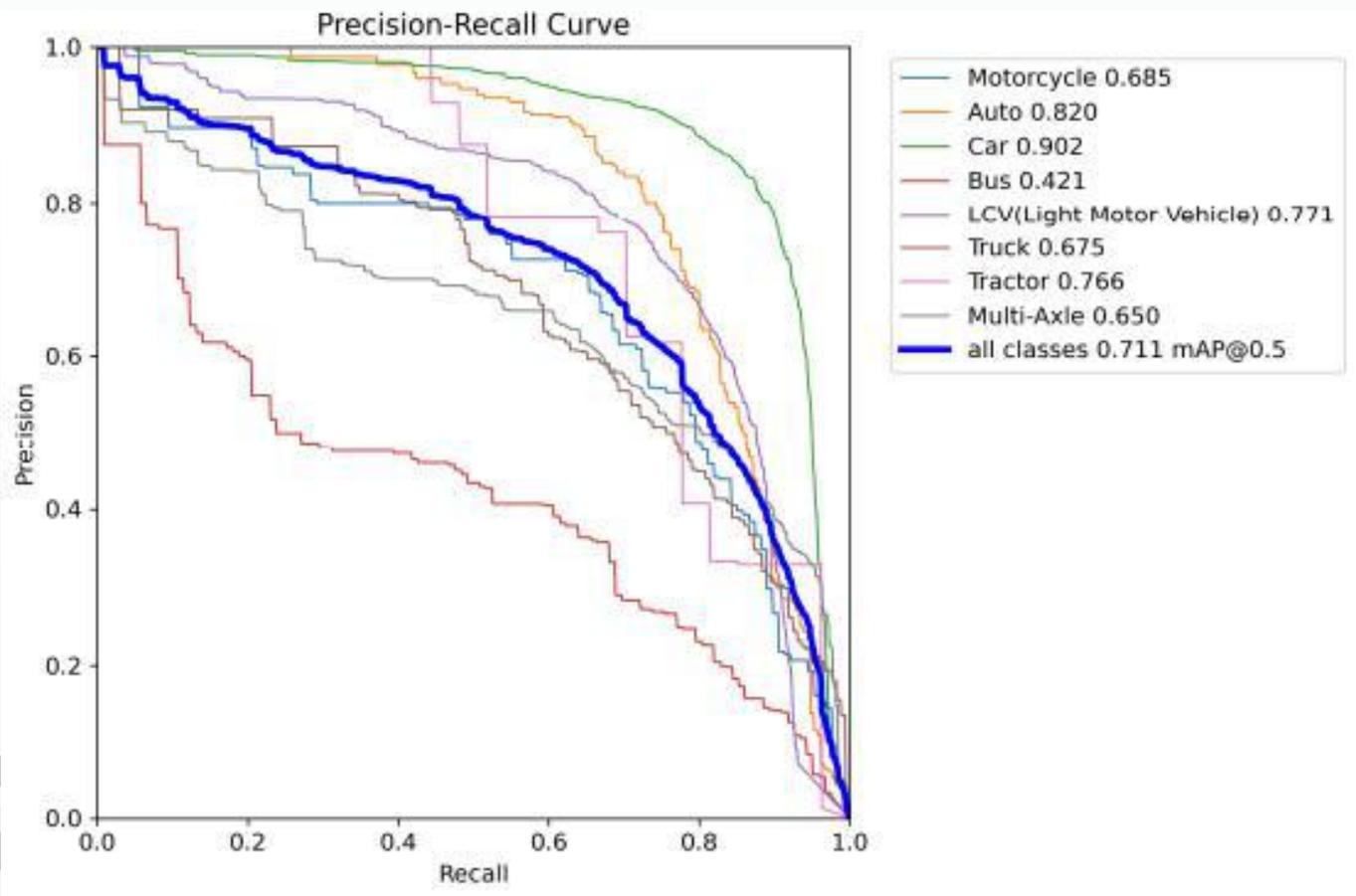


Рис. 8.1 – Крива Precision-Recall, що показує, як модель балансує між точністю та повнотою.

- Графік демонструє критичний баланс між точністю та повнотою.
- Найкращі результати: Клас Car досягає високих значень (0.902), підтверджуючи високу ефективність для найбільш представленого типу транспорту.
- Складні класи: Клас Bus залишається на рівні (0.421), що ідентифікує основну область для подальшого доопрацювання.



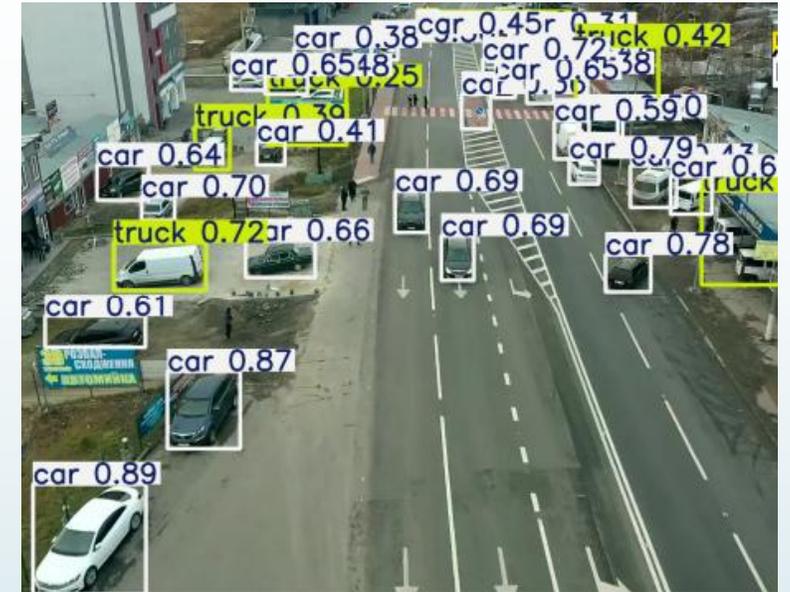
Рис.9.1 – Фрагмент відеокдру, використаний для експериментальних досліджень.

- Інтенсивний трафік: Кадр містить високу щільність дорожнього руху та різні категорії транспортних засобів (легкові, вантажні), що забезпечує об'єктивну оцінку класифікаційної здатності системи.
- Проблеми детекції: Наявність частково перекритих об'єктів є типовим викликом, який ускладнює точне визначення просторових меж та є прямою перевіркою ефективності модифікації.
- Зовнішні чинники: Природні умови освітлення, наявність тіней та елементів інфраструктури перевіряють стійкість алгоритмів до впливу зовнішніх чинників у реальних умовах експлуатації.

ВІЗУАЛЬНЕ ПОРІВНЯННЯ РЕЗУЛЬТАТІВ ВИЯВЛЕННЯ

Порівняння результатів виявлення на одному й тому ж складному кадрі підтверджує перевагу модифікованої моделі в точності.

Критерій	Базова YOLOv8 (Рис. 10.1)	Модифікована YOLOv8 (Рис. 10.2)
Кількість виявлень	35 (вища чутливість, ризик хибних спрацювань)	31 (менше виявлень, але вища точність)
Макс. Впевненість	0.89	0.92
Вантажні авто (Клас 7)	6 коректних виявлень	8 коректних виявлень



ВИСНОВКИ

1. Сформульовано задачу дослідження та вимоги до системи на підставі критичного аналізу сучасних методів YOLO та CNN.
2. Успішно розроблено та інтегровано спеціалізований модуль уваги (Attention Module) для модифікації архітектури YOLOv8.
3. Реалізовано навчання модифікованої моделі на датасеті та підготовлено об'єктивну систему метрик (*mAP*, *F1 - міра*) для оцінки.
4. Експериментально підтверджено покращення якості (точності) виявлення об'єктів у складних умовах порівняно з базовою моделлю YOLOv8.
5. Досягнуте підвищення точності критично важливе для стійкої роботи системи моніторингу дорожнього руху в реальних умовах.

АПРОБАЦІЯ:

1. Всеукраїнська науково-практична конференція «Актуальні проблеми кібербезпеки», на тему «Оптимізований комп'ютерний зір для ідентифікації транспорту в ІТС» ст. 208, 24 жовтня 2025 року.
2. III Всеукраїнській науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу», на тему «Комп'ютерний зір для ідентифікації та категоризації транспорту», що відбудеться 18 листопада 2025 року.

Дякую за увагу!