

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розроблення програмного модуля ідентифікації
номерних знаків автотранспортних засобів
(тема)

Виконав:

Здобувач 4 року навчання,
групи АКТАКІТ-21-3

Дмитро ГОЛОВАНЬ

(власне ім'я прізвище)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Автоматизація та
комп'ютерно-інтегровані технології

(повна назва освітньої програми)

Керівник доцент Рауф АЛЛАХВЕРАНОВ

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТАР

Ігор НЕВЛЮДОВ

(власне ім'я прізвище)

2025р.

Я, Головань Дмитро Сергійович, як здобувач вищої освіти ХНУРЕ, розумію та підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«12» червня 2025 р.

Дмитро ГОЛОВАНЬ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологій
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
Рівень вищої освіти перший (бакалаврський)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____
(підпис)

« 28 » квітня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Головань Дмитру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення програмного модуля ідентифікації номерних знаків автотранспортних засобів

затверджена наказом по університету від “ 19 ” травня 2025 р. № 390 Ст.

2. Термін подання здобувачем роботи “ 26 ” червня 2025 р.

3. Вихідні дані до роботи 3.1 Можливість виявлення транспортних засобів в режимі реального часу за допомогою відео та ідентифікація їхніх номерних знаків з високою точністю;

3.2 Забезпечення виявлення номерних знаків за різних умов освітлення, кутів та оклюзій;

3.3 Система ідентифікації номерних знаків – ALPR;

3.4 Бібліотеки – OpenCV, EasyOCR та Pandas;

3.5 Мова програмування – Python.

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ;

4.2 Аналіз існуючих методів розпізнавання номерних знаків автотранспортних засобів;

4.3 Складові компоненти для розроблення програмного модуля ідентифікації номерних знаків автотранспортних засобів;

4.4 Розроблення програмного модуля ідентифікації номерних знаків автотранспортних засобів;

4.5 Заходи і розрахунки для забезпечення безпечних умов праці;

4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 18 с. формату А4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Аналіз існуючих методів розпізнавання номерних знаків автотранспортних засобів</i>	28.04 – 04.05.25	<i>виконано</i>
2	<i>Складові компоненти для розроблення програмного модуля ідентифікації номерних знаків автотранспортних засобів</i>	05.05 – 14.05.25	<i>виконано</i>
3	<i>Розроблення програмного модуля ідентифікації номерних знаків автотранспортних засобів</i>	15.05 – 31.05.25	<i>виконано</i>
4	<i>Заходи і розрахунки для забезпечення безпечних умов праці</i>	01.06 – 10.06.25	<i>виконано</i>
5	<i>Оформлення пояснювальної записки</i>	11.06 – 14.06.25	<i>виконано</i>
6	<i>Подання роботи на перевірку Інтернет-системою StrikePlagiarism</i>	15.06 – 17.06.25	<i>виконано</i>
7	<i>Подання роботи на рецензію</i>	18.06 – 20.06.25	<i>виконано</i>
8	<i>Подання роботи на підпис зав. кафедри</i>	21.06 – 23.06.25	<i>виконано</i>
9	<i>Подання кваліфікаційної роботи в ЕК</i>	24.06.25	<i>виконано</i>

Дата видачі завдання 28.04.2025 р.

Здобувач _____
(підпис)

Дмитро ГОЛОВАНЬ

Керівник роботи _____
(підпис)

доцент Рауф АЛЛАХВЕРАНОВ
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 67 с., 1 табл., 19 рис., 3 дод., 24 джерел.

ІДЕНТИФІКАЦІЯ, ПРОГРАМНИЙ МОДУЛЬ, НОМЕРНІ ЗНАКИ, АВТОТРАНСПОРТНІ ЗАСОБИ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, КОМП'ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ.

Мета роботи – розробка програмного модуля розпізнавання та ідентифікації номерних знаків автотранспортних засобів, яка базується на сучасних методах комп'ютерного зору та машинного навчання.

Об'єкт розробки – розпізнавання та ідентифікації номерних знаків автотранспортних засобів.

Предмет розробки – програмний модуль розпізнавання та ідентифікації номерних знаків автотранспортних засобів.

У кваліфікаційній роботі проведено розробка програмного модулю розпізнавання та ідентифікації номерних знаків автотранспортних засобів на основі сучасних методів комп'ютерного зору та машинного навчання. Розробка програмного модуля включала кілька ключових етапів: аналіз існуючих технологій, вибір оптимальної моделі для розпізнавання автомобілів (YOLOv8n) та алгоритму відстежування (SORT), навчання моделі, та інтеграцію бібліотеки EasyOCR для розпізнавання та ідентифікації номерних знаків автотранспортних засобів. Програмний модуль був реалізований як консольний додаток, здатний обробляти відео з транспортними потоками та видавати відео з анотованими номерними знаками.

Отримані результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», зокрема до пункту 9.4 «Розвиток високотехнологічного машинобудування».

ABSTRACT

Explanatory note: 67 pp., 1 tab., 19 figs., 3 appendices, 24 sources.

IDENTIFICATION, SOFTWARE MODULE, LICENSE PLATES, MOTOR VEHICLES, SOFTWARE, COMPUTER VISION, MACHINE LEARNING.

The purpose of the work is to develop a software module for recognizing and identifying vehicle license plates, which is based on modern methods of computer vision and machine learning.

The object of development is the recognition and identification of vehicle license plates.

The subject of development is the software module for recognizing and identifying vehicle license plates.

In the qualification work, a software module for recognizing and identifying vehicle license plates was developed based on modern computer vision and machine learning methods. The development of the software module included several key stages: analysis of existing technologies, selection of the optimal model for vehicle recognition (YOLOv8n) and tracking algorithm (SORT), model training, and integration of the EasyOCR library for recognizing and identifying vehicle license plates. The software module was implemented as a console application capable of processing video with traffic flows and outputting video with annotated license plates.

The results of the work can be attributed to Sustainable Development Goal 9 “Industry, Innovation and Infrastructure”, in particular to paragraph 9.4 “Development of high-tech engineering”.

ЗМІСТ

Перелік скорочень	9
Вступ... ..	10
1 Аналіз існуючих методів розпізнавання номерних знаків автотранспортних засобів	12
1.1 Технології розпізнавання номерних знаків	12
1.2 Аналіз сучасних підходів та методів	15
1.3 Висновки до розділу	23
2 Складові компоненти для розробки модуля ідентифікації номерних знаків автотранспортних засобів	25
2.1 Мови програмування для розробки систем LPR	25
2.2 Бібліотека OpenCV та її можливості	29
2.3 Функціонал моделі YOLOv8 та побудова датасетів	32
2.4 Висновки до розділу	39
3 Розробка програмного модуля ідентифікації номерних знаків автотранспортних засобів	40
3.1 Вибір відповідної моделі виявлення об'єктів	40
3.2 Вибір алгоритму відстеження об'єктів	42
3.3 Реалізація моделі ідентифікації об'єктів	43
3.4 Огляд проблеми розпізнавання об'єктів	49
3.5 Важливість інтерполяції даних для належної візуалізації результатів .	51
3.6 Перевірка та аналіз роботи програмного модуля	53
3.7 Висновки до розділу	55
4 Заходи і розрахунки для забезпечення безпечних умов праці	57
4.1 Аналіз умов праці на робочому місці	57
4.2 Промислова безпека на робочому місці	57
4.3 Виробнича санітарія у приміщенні	58
4.4 Пожежна безпека виробничого приміщення	60

Висновки	63
Перелік джерел посилання	65
Додаток А Результати тренування моделі	68
Додаток Б Лістинг програми	69
Додаток В Демонстраційний матеріал	70

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних;

ЕОМ – електронно-обчислювальна машина;

КЗ – комп'ютерний зір;

КПО – коефіцієнт природної освітленості;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

ТЗ – технічне завдання.

ВСТУП

У сьогоднішній бурхливий розвиток інформаційних технологій суттєво впливає на безліч галузей діяльності людини, зокрема, особливе місце належить транспортній галузі. Так, збільшення кількості транспортних засобів сприяє впровадженню ефективних систем для контролю, керування й автоматизації процесів, які безпосередньо пов'язані з дорожнім рухом. Серед таких систем виокремимо систему ідентифікація номерних знаків автотранспортних засобів ALPR (Automatic License Plate Recognition), яка набуває активного застосування в різних напрямках, як-от від забезпечення громадської безпеки до автоматизації платних доріг та паркінгів.

Власне система ідентифікація номерних знаків автотранспортних засобів ґрунтується як на методах комп'ютерного зору, так і машинного навчання, що, зі свого боку, дозволяють автоматично виявляти, сегментувати й інтерпретувати ці знаки на зображеннях. Застосування цих методів забезпечує високу точність і швидкість ідентифікація. Отже, такі системи стають незамінними в реаліях сучасного міського середовища.

Мета роботи – розробка програмного модуля розпізнавання та ідентифікації номерних знаків автотранспортних засобів, яка базується на сучасних методах комп'ютерного зору та машинного навчання.

Об'єкт розробки – розпізнавання та ідентифікації номерних знаків автотранспортних засобів.

Предмет розробки – програмний модуль розпізнавання та ідентифікації номерних знаків автотранспортних засобів.

Важливість такого дослідження обумовлена необхідністю у підвищенні ефективності автоматизації транспортних процесів, а також покращенні рівня безпеки дорожнього руху. У цілому результати роботи можуть стати актуальними як для створення комерційних продуктів, так і для розвитку сучасних інтелектуальних транспортних систем.

У межах даної роботи буде здійснено:

- аналіз існуючих підходів до розпізнавання та ідентифікації номерних знаків автотранспортних засобів;
- розробка алгоритмів для обробки зображень і ідентифікації символів номерних знаків автотранспортних засобів;
- експериментальна перевірка ефективності запропонованих рішень;
- проведення заходів і розрахунків для забезпечення умов безпечної праці в лабораторії, де проводились дослідження з кваліфікаційної роботи.

Пояснювальну записку кваліфікаційної роботи оформлено згідно з ДСТУ 3008:2015 [1] та з рекомендаціями з підготовки і оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти [2-3], отримані результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», зокрема до пункту 9.4 «Розвиток високотехнологічного машинобудування».

1 АНАЛІЗ ІНСНУЮЧИХ МЕТОДІВ ІДЕНТИФІКАЦІЇ НОМЕРНИХ ЗНАКІВ АВТОТРАНСПОРТНИХ ЗАСОБІВ

1.1 Технології ідентифікації номерних знаків автотранспортних засобів

Ідентифікації номерних знаків (LPR) автотранспортних засобів, також відоме як автоматичне ідентифікації номерних знаків (ANPR), являє собою технологію, що дозволяє знімати відео або зображення з номерних знаків і перетворювати оптичні дані в цифрову інформацію в режимі реального часу [5]. Ця технологія широко використовується для операцій керування транспортними засобами, таких як безквиткове паркування, оплата за проїзд, інтелектуальні транспортні системи (ITS), виявлення викрадених автомобілів, інтелектуальне виставлення рахунків та багато інших програм, особливо в Європі, і набуває популярності в Америці.

1.1.1 Опис можливостей та застосування систем ідентифікації номерних знаків (ALPR) у сучасному світі

Додаючи цифрову інформацію про номерний знак разом з іншими даними (напрямок і швидкість автомобіля), ідентифікації номерних знаків дозволяє операторам збирати та аналізувати більше даних про кожен транспортний засіб, що проїжджає через контрольну точку. Це включає інформацію про час, місце, напрямок і швидкість транспортного засобу, його походження, можливі обмеження або сповіщення безпеки, а також дані про водія [6].

Застосування в різних сферах:

– безпека: поліцейські сили використовують ідентифікації номерних знаків для виявлення викрадених, розшукуваних або незастрахованих транспортних засобів за допомогою вбудованих камер на своїх автомобілях або стаціонарних контрольних точках;

– паркінг: LPR дозволяє автоматизувати та оцифрувати операції паркування, що створює багато можливостей для покращення обслуговування клієнтів. Системи безквиткового паркування забезпечують безконтактне та швидке паркування без зупинки біля шлагбаумів;

– розумні міста: міста використовують ідентифікації номерних знаків для розробки кращої політики на основі глибшого розуміння дорожнього руху;

– контроль доступу автотранспортних засобів: резиденції, компанії та уряди застосовують LPR для автоматизації та оптимізації керування доступом до транспортних засобів;

– толінг та ІТС: ідентифікації номерних знаків сприяє розробці нових рішень для оплати дорожнього руху та ІТС. Камери LPR інтегруються до існуючих систем, покращуючи моніторинг транспортних засобів і можливості визначення місця розташування, такі як Toll-By-Plate.

Системи ідентифікації номерних знаків приносять значні переваги як компаніям, так і їхнім клієнтам. Вони дозволяють зменшити витрати, збільшити доходи та покращити досвід клієнтів завдяки швидшій, безконтактній службі паркування. Завдяки цим технологіям уряди та компанії можуть створювати більш розумні та ефективні політики, забезпечувати краще обслуговування клієнтів.

1.1.2 Визначення основних компонентів системи ALPR

Системи LPR складаються з наступних етапів [7]:

- отримання зображення;
- попередня обробка;
- локалізація номерного знака;
- сегментація символів;
- виділення ознак;
- ідентифікації символів і пост обробка.

Отримання зображень. У LPR отримання зображення – це процес захоплення зображення номерного знака. Це можна зробити за допомогою різноманітних методів, зокрема камер, встановлених на транспортних засобах чи будівлях, або портативних пристроїв. Існує два основних типи камер, що використовуються в LPR:

- інфрачервоні камери призначені для роботи в умовах слабого освітлення і можуть знімати номерні знаки, навіть якщо вони не освітлені;
- камери видимого світла використовуються в добре освітлених місцях і можуть захоплювати більше деталей, ніж інфрачервоні камери. Камери, які використовуються в системах LPR, повинні відповідати певним вимогам, щоб досягти високих показників точності. Камера повинна мати високу роздільну здатність, щоб вона могла зафіксувати всі деталі номерного знака та створювати чітке зображення незалежно від умов освітлення.

Попередня обробка. Це перша фаза LPR, яка передбачає підготовку зображення до обробки. На цьому кроці зображення камери налаштовуються для зменшення шуму та покращення контрастності.

Локалізація номерних знаків. На другому етапі системи LPR використовують алгоритми для ідентифікації номерних знаків на зображенні. Це передбачає ідентифікації форми, розміру та розташування номерного знака та видалення будь-якого залишкового фонового шуму чи інших об'єктів, які не є частиною номерного знака.

Сегментація символів. На третьому етапі системи LPR використовують алгоритми для сегментації символів у номерному знаку. Цей крок призначений для того, щоб система LPR могла точно розпізнавати символи на зображенні номерного знака.

Виділення ознак. На цьому кроці алгоритми LPR витягують особливості із зображення номерного знака. Це передбачає аналіз форми, розміру та орієнтації кожного символу та порівняння з базою даних відомих особливостей.

Ідентифікації символів. На завершальному етапі алгоритми LPR порівнюють витягнуті характеристики з базою даних відомих номерних знаків, щоб ідентифікувати та розпізнати номерний знак.

Подальша обробка. У LPR постобробка зображення є останнім кроком у процесі ідентифікації номерних знаків. Цей крок передбачає аналіз результатів системи LPR і вжиття відповідних дій на основі результатів. Постобробка може включати різноманітні завдання, такі як надсилання сповіщень, коли номерний знак розпізнається, збереження інформації про номерний знак у базі даних або маршрут транспортного засобу до певного місця. Постобробка є важливим кроком у LPR, оскільки вона дозволяє налаштовувати системи LPR відповідно до конкретних потреб кожної організації. Налаштувавши параметри постобробки, системи LPR можна налаштувати відповідно до конкретних вимог кожної програми.

1.2 Аналіз сучасних підходів та методів

1.2.1 Розгляд методів обробки зображень для виявлення та сегментації об'єктів

Традиційні методи обробки зображень для виявлення та сегментації об'єктів використовують базові алгоритми комп'ютерного зору, які працюють з низькорівневими характеристиками зображення [8]. Основні методи включають [9]:

а) порогове перетворення (thresholding) є одним з найпростіших і найпоширеніших методів обробки зображень (рис. 1.1). Воно використовується для сегментації зображення на основі інтенсивності пікселів. Цей метод полягає у встановленні порогу, за яким пікселі поділяються на два класи (належать об'єкту (номерному знаку) та належать фону):

1) метод глобального порогового перетворення. Встановлюється єдиний поріг для всього зображення. Пікселі з інтенсивністю вище порогу відносяться до об'єкту, нижче порогу – до фону;

2) метод адаптивного порогового перетворення. Використовуються різні пороги для різних ділянок зображення, що дозволяє краще враховувати локальні зміни освітлення. Порогове значення обчислюється для кожного пікселя на основі середнього або медіанного значення інтенсивностей у сусідніх областях;



Рисунок 1.1 – Приклад порогового та адаптивного перетворення

б) виділення контурів (edge detection) є ключовим етапом у виявленні та сегментації об'єктів. Контури дозволяють визначити межі об'єктів на зображенні (рис. 1.2):

1) оператор Собеля: обчислює градієнт інтенсивності зображення в горизонтальному та вертикальному напрямках. Контури визначаються там, де градієнт найбільший;

2) оператор Кенні: більш складний метод, який включає кроки згладжування зображення, обчислення градієнта, не-максимального приглушення та подвійного порогового перетворення. Він дозволяє виявити тонкі та точні контури;



Рисунок 1.2 – Виділення контурів оператором Кенні

в) морфологічні операції використовуються для подальшої обробки бінарних зображень після порогового перетворення і виділення контурів. Вони допомагають видаляти шуми, заповнювати прогалини та покращувати якість сегментованих об'єктів (рис. 1.3). Перелічимо основні морфологічні операції:

- 1) ерозія – видаляє пікселі на межах об'єктів. Застосовується для зменшення розмірів об'єктів і видалення дрібних шумів;
- 2) дилатація – додає пікселі до меж об'єктів. Використовується для збільшення розмірів об'єктів і заповнення прогалин;
- 3) розкриття: комбінація ерозії та дилатації. Видаляє дрібні об'єкти та шуми, зберігаючи розмір і форму великих об'єктів;
- 4) закриття: комбінація дилатації та ерозії. Заповнює дрібні прогалини в об'єктах, зберігаючи їх загальну форму.



Рисунок 1.3 – Зниження шуму зображення морфологічними операціями

В ALPR-системах ці методи обробки зображень використовуються послідовно для виявлення номерних знаків на зображеннях та їх сегментації:

- порогове перетворення виділяє області, що можуть відповідати номерним знакам;
- виділення контурів допомагає точно визначити межі цих областей;
- морфологічні операції покращують якість виявлених областей, видаляючи шуми та заповнюючи прогалини, що дозволяє ефективніше сегментувати символи номерного знака.

Комбінація цих традиційних методів дозволяє досягати високої точності та надійності в системах ідентифікації номерних знаків, особливо в умовах доброї якості зображень та однорідного фону.

1.2.2 Алгоритми машинного та глибокого навчання, які використовуються для ідентифікації номерних знаків

Метод k -найближчих сусідів (k -nearest neighbors) є простим, але ефективним підходом до класифікації. Він базується на тому, що зображення символів сортується за класами і записуються в базу даних. Потім вводиться міра відстані між цими зображеннями. Під час ідентифікації обраний символ порівнюється з усіма символами в базі, і для визначення його класу розглядаються k найближчих сусідів [10]. Проте, виникає проблема швидкості розрахунку відстані між даними, особливо при бінаризації. Цю проблему можна вирішити, маючи велику базу символів, зафіксованих у різних умовах. Головною перевагою методу є його простота, що дозволяє легко налаштувати алгоритм для отримання потрібних результатів.

OCR – оптичне ідентифікації символів, є одним із основних методів ідентифікації тексту в зображеннях. Він передбачає виявлення текстових областей на зображеннях, сегментацію окремих символів і їх ідентифікації за допомогою ідентифікації образів або алгоритмів машинного навчання [7]. Класичні методи оптичного ідентифікації символів включають такі методи, як зіставлення шаблонів, вилучення ознак (наприклад, гістограма

орієнтованих градієнтів – HOG) і класифікатори, такі як опорні векторні машини (SVM) або k -найближчі сусіди (k -NN).

Mask Region-based Convolutional Neural Network (Mask R-CNN), є розширенням фреймворку Faster R-CNN з архітектурою зображеною на рис. 1.4.

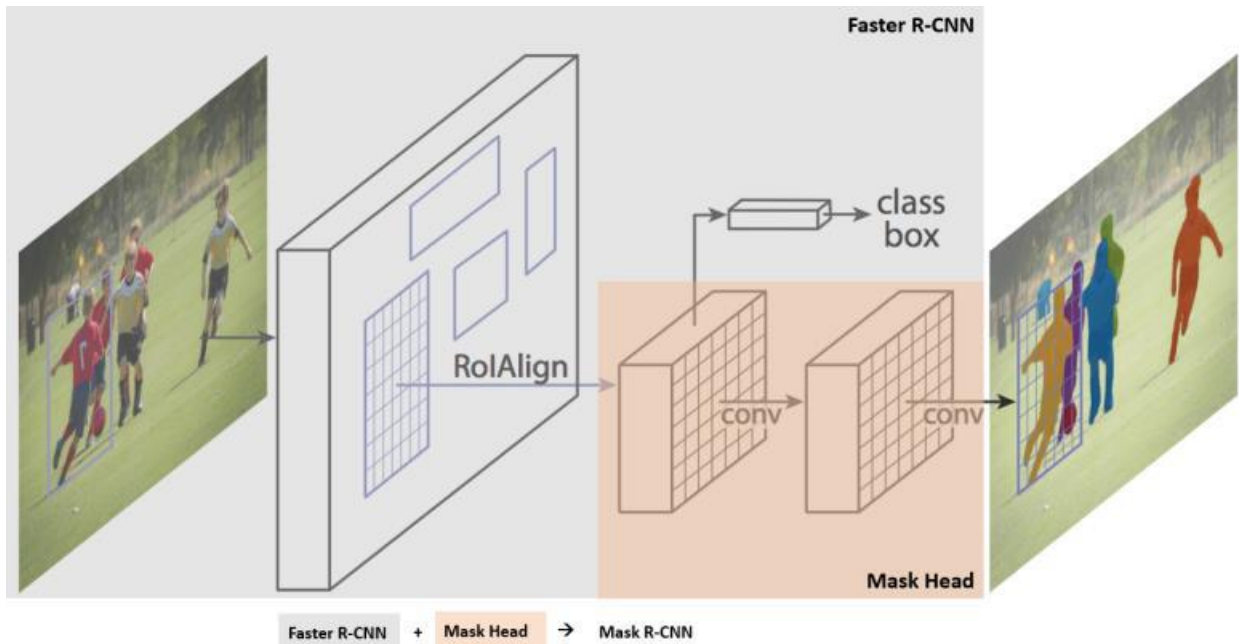


Рисунок 1.4 – Архітектура Mask R-CNN

Mask R-CNN в основному використовується для задач виявлення об'єктів в комп'ютерному зорі [12]. Основна ідея його полягає в тому, щоб розширити Faster R-CNN, додавши додаткову гілку для прогнозування масок сегментації поряд із існуючими гілками для виявлення об'єктів. Розберемо принцип роботи Mask R-CNN:

– магістральна мережа: як і Faster R-CNN, Mask R-CNN починається з магістральної згорткової нейронної мережі (CNN), такої як ResNet або VGG, для вилучення карт функцій із вхідного зображення. Ці карти функцій фіксують ієрархічне представлення зображення, де нижні рівні містять функції низького рівня (наприклад, краї, текстури), а вищі рівні фіксують більш абстрактні характеристики (наприклад, частини об'єкта, форми);

– регіональна мережа пропозицій (RPN): подібно до Faster R-CNN, Mask R-CNN використовує мережу регіональних пропозицій (RPN) для створення обмежувальних рамок-кандидатів (пропозицій регіонів), які можуть містити об'єкти. Ці пропозиції генеруються на основі опорних блоків у різних масштабах і пропорціях і вдосконалюються, щоб краще відповідати об'єктам під час навчання;

– регіон інтересу (RoI): після того, як RPN генерує пропозиції щодо регіону, Mask R-CNN використовує RoI Align для отримання карт функцій фіксованого розміру з магістральної мережі для кожної пропозиції. RoI Align усуває проблеми з розбіжністю, пов'язані з оригінальним методом об'єднання RoI, забезпечуючи точне просторове вирівнювання між вхідним зображенням і витягнутими об'єктами;

– класифікація та регресія обмежувальної рамки: карти функцій, вирівняні за RoI, подаються в дві однорідні повністю пов'язані гілки: одну для класифікації об'єктів, а іншу – для регресії обмежувальної рамки. Гілка класифікації передбачає ймовірність належності кожної пропозиції до попередньо визначених класів об'єктів, тоді як гілка регресії уточнює координати обмежувальної рамки пропозицій, щоб краще відповідати об'єктам;

– передбачення маски: на додаток до класифікації та регресії обмежувальної рамки, Mask R-CNN представляє додаткову гілку для прогнозування масок сегментації. Ця гілка приймає функції, узгоджені з RoI, як вхідні дані та виводить двійкову маску для кожної пропозиції незалежно від класу. Кожна маска передбачає попіксельну сегментацію об'єкта в пропозиції;

– функція втрати: під час навчання Mask R-CNN оптимізує багатозадачну функцію втрати, яка поєднує три компоненти: втрата класифікації (наприклад, втрата крос-ентропії) для класифікації об'єктів, втрати регресії обмежувальної рамки (наприклад, плавні втрати L1) для

уточнення координат обмежувальної рамки, втрата маски (наприклад, двійкова втрата перехресної ентропії) для масок попіксельної сегментації.

Завдяки спільній оптимізації цих компонентів Mask R-CNN вчиться виявляти об'єкти, уточнювати їх обмежувальні рамки та точно сегментувати їх на зображенні. Остаточний вихід складається з виявлених обмежувальних рамок, міток класів і масок сегментації для кожного екземпляра об'єкта у вхідному зображенні.

1.2.3 Огляд відомих технологічних рішень для ідентифікації номерних знаків

Проаналізувавши джерела [9-11], можна зробити висновки, що системи LPR зазвичай використовуються в галузі правоохоронних органів, зборі мита та організації паркування. З огляду на це були обрані, для аналізу, три популярні комплексні системи LPR:

a) Vigilant Solutions може похвалитися високою точністю і надійністю завдяки передовим методам попередньої обробки та машинного навчання та високою ефективністю у різноманітних умовах навколишнього середовища:

1) попередня обробка зображень включає в себе налаштування контрастності, зменшення шуму та нормалізацію зображення для покращення чіткості номерних знаків;

2) використовує алгоритми виявлення контурів (наприклад, Canny або Sobel) і морфологічні операції, такі як розкриття та ерозія, для визначення меж номерних знаків на зображенні;

3) коли номерний знак локалізовано, використовується метод OCR для ідентифікації буквено-цифрових символів;

4) інтегрує моделі машинного навчання, навчені великими наборами даних, щоб підвищити точність і надійність виявлення номерних знаків і ідентифікації символів, особливо за різного освітлення та погодних умов;

б) Genetec AutoVu має високу точність виявлення та зчитування номерних знаків завдяки використанню глибокого навчання та передового OCR. Інтегрується з іншими системами безпеки для комплексного управління. Але вимагає значних обчислювальних ресурсів для моделей глибокого навчання та має високу початкову вартість і складність у налаштуванні. AutoVu використовує:

1) згорточні нейронні мережі (CNN) для виявлення номерних знаків на зображенні. Ці мережі навчені на величезній кількості даних, щоб розпізнавати особливості номерних знаків;

2) алгоритми сегментації для виділення окремих символів. Це часто передбачає такі методи, як аналіз зв'язаних компонентів, щоб відрізнити персонажів від фону;

3) найсучасніші методи OCR, включаючи повторювані нейронні мережі (RNN) і мережі довготривалої короткочасної пам'яті (LSTM), які особливо ефективні для послідовних даних, таких як текст;

4) методи покращення зображення, такі як адаптивне вирівнювання гістограми та зменшення шумів, використовуються для покращення якості зображення, особливо в умовах слабого освітлення;

в) ELSAG Plate Hunter – система, яка розроблена для захоплення та обробки зображень у реальному часі з використанням оптимізованих алгоритмів для забезпечення мінімальної затримки між виявленням і ідентифікаціям. ELSAG Plate Hunter гарантує надійну продуктивність у програмах реального часу завдяки оптимізованим алгоритмам але може бути не настільки гнучким або точним, як системи на основі глибокого навчання в дуже різноманітних умовах. Ця система використовує:

1) традиційні методи обробки зображень для виявлення номерних знаків, включаючи алгоритми виявлення країв і перетворення Хафа для визначення прямокутної форми знака;

2) метод відповідності шаблону для зіставлення виявленої області номерного знака з попередньо визначеним шаблоном форм і розмірів номерних знаків, забезпечуючи точну локалізацію;

3) гібридний підхід, що поєднує традиційні методи оптичного ідентифікації тексту з удосконаленнями машинного навчання. Це забезпечує надійне ідентифікації символів навіть у різних умовах.

1.3 Висновки до розділу

У даному розділі кваліфікаційної роботи були оглянуті складні методи комп'ютерного зору та алгоритми для покращення ідентифікації транспортних засобів та обробки даних. Ідентифікації автомобільних номерних знаків (LPR), також відоме як автоматичне ідентифікації номерних знаків (ALPR), є життєво важливим у сучасних інтелектуальних транспортних системах, значно покращуючи безпеку, керування паркуванням, платні дороги та контроль доступу транспортних засобів.

Системи LPR передбачають захоплення зображень високої роздільної здатності за допомогою камер інфрачервоного та/або видимого світла, попередню обробку для зменшення шуму та підвищення контрастності, а також локалізацію номерного знака. Потім система сегментує табличку на символи, виділяє ознаки та розпізнає ці символи в базі даних.

Важливу роль відіграють методи обробки зображень, на яких базуються системи LPR. Порогове визначення – це процедура, за якої пікселі поділяються на ті, що складають об'єкт, і ті, що складають фон, на основі інтенсивності пікселів; алгоритмів передбачає глобальні процедури для незмінного освітлення та адаптивні процедури, які спрямовані на підвищення точності за умов зміни рівня освітлення у різних областях зображення. Виявлення країв використовується, щоб зробити межі об'єктів чіткими. Морфологічні операції підвищують якість зображення, усуваючи шум і

заповнюючи проміжки від попередньо зроблених операцій за допомогою ерозії, розширення, розкриття та закриття.

Розширене машинне навчання та алгоритми глибокого навчання є ключовими в сучасних системах LPR. Алгоритм k-найближчих сусідів (k-NN) пропонує просту класифікацію на основі близькості ознак. Оптичне ідентифікації символів (OCR) використовує традиційні методи, такі як зіставлення шаблонів і виділення ознак у поєднанні з такими класифікаторами, як Support Vector Machines (SVM) і k-NN. Mask R-CNN, розширений підхід глибокого навчання, забезпечує точне виявлення об'єктів і сегментацію, розширюючи Faster R-CNN додатковою гілкою передбачення по масці.

Такі технологічні рішення, як Vigilant Solutions, Genetec AutoVu та ELSAG Plate Hunter, використовують ці методи для забезпечення високої точності та продуктивності в реальному часі в різних середовищах. Ці системи зменшують операційні витрати, збільшують дохід і покращують взаємодію з клієнтами завдяки швидшим безконтактним послугам. Зрештою, технології LPR дозволяють розумніше й ефективніше керувати транспортом і надавати послуги, сприяючи розвитку розумних міст.

2 СКЛАДОВІ КОМПОНЕНТИ ДЛЯ РОЗРОБЛЕННЯ ПРОГРАМНОГО МОДУЛЯ ІДЕНТИФІКАЦІЇ НОМЕРНИХ ЗНАКІВ АВТОТРАНСПОРТНИХ ЗАСОБІВ

2.1 Вибір мови програмування для розробки систем LPR

2.1.1 Мова програмування Python

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією. Розроблена на початку 1990-х років Гвідо ван Россумом [12].

Python – мультипарадигмальна високорівнева мова програмування загального призначення з динамічною суворою типізацією та автоматичним керуванням пам'яттю, орієнтована на підвищення продуктивності розробника, читабельності коду та його якості, а також на забезпечення переносності написаних нею програм.

Переваги мови програмування Python [13]:

- бібліотеки та інструменти: Python має широкий набір бібліотек для обробки зображень (OpenCV), машинного навчання (scikit-learn), і глибокого навчання (TensorFlow, PyTorch, Keras). Це дозволяє швидко прототипувати та реалізовувати складні алгоритми;

- простота у використанні: Python є інтуїтивно зрозумілим і простим у використанні, що прискорює процес розробки та тестування. Його синтаксис легкий для вивчення, особливо для новачків;

- широка підтримка: велика спільнота розробників забезпечує доступ до багатьох ресурсів, включаючи документацію, форуми та приклади коду. Це сприяє швидкому вирішенню проблем, які можуть виникнути під час розробки;

- портативність: Python є міжплатформною мовою, що дозволяє запускати код на різних операційних системах без значних змін.

Недоліки мови програмування Python [13]:

- продуктивність: Python є інтерпретованою мовою, тому його продуктивність може бути нижчою порівняно з компільованими мовами, такими як C++;
- використання ресурсів: Python може споживати більше пам'яті та обчислювальних ресурсів, що може бути критичним для додатків;
- потребують високої продуктивності.

2.1.2 Мова програмування C++

C++ – універсальна мова програмування високого рівня з підтримкою декількох парадигм програмування. Зокрема: об'єктно-орієнтованої та процедурної. Розроблена Б'ярном Страуструпом (англ. Bjarne Stroustrup) в AT&T Bell Laboratories (Мюррей-Хілл, Нью-Джерсі) у 1979 році.

Синтаксис C++ успадкований від мови C. Від початку одним із принципів розроблення було збереження сумісності з Cі. Проте, C++ не є в строгому сенсі надмножиною C: безліч програм, які можуть однаково успішно транслюватися як компіляторами C, так і компіляторами C++, доволі велика, але не охоплює всіх можливих програм на C.

C++ – компільовальна, статично типізована мова програмування загального призначення.

Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування. Мова має багату стандартну бібліотеку, яка містить поширені контейнери та алгоритми, введення/виведення, регулярні вирази, підтримку багатопоточності та інші можливості. C++ поєднує властивості як високорівневих, так і низькорівневих мов. Порівняно з її попередником – мовою C – найбільшу увагу приділено підтримці об'єктно-орієнтованого та узагальненого програмування.

C++ широко використовується для розробки програмного забезпечення, будучи однією з найпопулярніших мов програмування. Сфера

її застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для вбудованих систем, високопродуктивних серверів, а також комп'ютерних ігор. Існує безліч реалізацій мови C++, як безкоштовних, так і комерційних і для різних платформ [14]. Наприклад, на платформі x86 це GCC, Clang, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder та інші. C++ справив величезний вплив на інші мови програмування, насамперед на Java і C#.

Переваги мови програмування C++ [14]:

- продуктивність: C++ є компільованою мовою, що забезпечує високу швидкість виконання та ефективність використання апаратних ресурсів. Це важливо для реального часу обробки відео та зображень;

- контроль над ресурсами: C++ надає низькорівневий доступ до пам'яті та апаратних ресурсів, що дозволяє оптимізувати додатки для максимальної продуктивності;

- бібліотеки: Існує багато бібліотек для обробки зображень (OpenCV) та інтеграції з глибоким навчанням (TensorFlow, Caffe). Це дозволяє використовувати потужні інструменти для створення високопродуктивних систем.

Недоліки мови програмування C++ [14]:

- складність у використанні: C++ є складнішою у вивченні та використанні мовою порівняно з Python. Вона вимагає більше часу на розробку та налагодження коду;

- менша підтримка в галузі глибокого навчання: Хоча C++ має бібліотеки для глибокого навчання, вони не так широко використовуються, як аналогічні бібліотеки для Python.

2.1.3 Мова програмування Java

Java – об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java [15].

Із середини 1990-х років мову почали широко використовувати для написання клієнтських додатків і серверного програмного забезпечення. Тоді ж певного поширення набула технологія Java-апплетів – графічних Java-додатків, що вбудовуються у вебсторінки; з розвитком можливостей динамічних веб-сторінок у 2000-ті роки технологія стала застосовуватися рідко.

З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems».

Переваги мови програмування Java [15]:

- портативність: Java є міжплатформною мовою, що забезпечує високу портативність завдяки Java Virtual Machine (JVM). Це дозволяє запускати код на будь-якій платформі, де встановлена JVM;

- підтримка підприємств: Java широко використовується в корпоративних додатках, що забезпечує стабільність та надійність розроблених систем;

- бібліотеки: існує велика кількість бібліотек для обробки зображень (JavaCV) та машинного навчання (Weka, Deeplearning4j), що дозволяє реалізувати складні алгоритми.

Недоліки мови програмування Java [15]:

- продуктивність: продуктивність Java може бути нижчою порівняно з C++, хоча зазвичай вона є вищою, ніж у Python, завдяки компіляції у байт-код та оптимізаціям JVM;

- менша популярність у сфері глибокого навчання: Java не так широко використовується для глибокого навчання, як Python. Це означає меншу кількість бібліотек та меншу підтримку спільноти в цій галузі.

Для розробки модуля ідентифікації номерних знаків найбільш оптимальною мовою програмування є Python. Цей вибір обумовлений наступними ключовими перевагами:

- широкий набір бібліотек та інструментів для обробки зображень (OpenCV), машинного навчання (scikit-learn) та глибокого навчання

(TensorFlow, PyTorch, Keras), що спрощує розробку та інтеграцію різних компонентів системи;

- простота у використанні та навчанні, що дозволяє швидше реалізовувати нові ідеї та алгоритми;

- велика підтримка спільноти та наявність численних ресурсів для навчання, що забезпечує швидке вирішення проблем та обмін досвідом.

Таким чином, Python забезпечує ефективність, гнучкість та швидкість розробки, що робить його ідеальним вибором для створення надійної системи ідентифікації автономних знаків.

2.2 Бібліотека OpenCV та її можливості

2.2.1 Опис бібліотеки OpenCV

OpenCV (Open Source Computer Vision Library) – це програмне забезпечення з відкритим вихідним кодом для комп'ютерного зору та машинного навчання, спочатку розроблене Intel, а потім підтримуване Willow Garage та Itseez (яке пізніше було придбано Intel). Розроблений, щоб забезпечити загальну інфраструктуру для програм комп'ютерного зору та прискорити використання машинного сприйняття в комерційних продуктах, OpenCV містить понад 2500 оптимізованих алгоритмів для виконання широкого спектру завдань у сфері комп'ютерного зору [16].

OpenCV був створений у 1999 році та з моменту першого випуску використовувався за ліцензією BSD для академічного та комерційного вживання. З моменту випуску цієї першої версії було досягнуто значного прогресу в цій бібліотеці, яка виросла далеко за межі бібліотек базової обробки зображень до бібліотек машинного навчання та тепер підтримує інфраструктуру глибокого навчання. Завдяки цьому OpenCV став корисним у багатьох задачах комп'ютерного зору.

Бібліотека доступна для Windows, Linux, macOS, iOS, Android і має реалізацію на C++, Python, Java та Matlab, що розширює можливості її використання серед розробників з різними мовними уподобаннями.

Основні функції OpenCV поширюються на наступний широкий спектр завдань: обробка зображень, фільтрація, перетворення простору кольорів, геометричні перетворення, а в аналізі відео – такі методи, як виявлення руху, відстеження об'єктів і віднімання фону [17]. Він включає в себе надійні алгоритми для виявлення об'єктів, такі як ідентифікації обличчя і пішоходів, і інтегрує алгоритми машинного навчання, наприклад, опорні векторні машини, нейронні мережі та k-найближчі сусіди. OpenCV також взаємодіє з популярними бібліотеками глибокого навчання, такими як TensorFlow, Torch/PyTorch і Caffe, таким чином маючи можливість реалізувати моделі глибокого навчання.

OpenCV розроблено для забезпечення функціональності в реальному часі. Алгоритми в бібліотеці оптимізовано для забезпечення дуже швидкого виконання. Він також використовує апаратне прискорення за допомогою GPU-обчислень із CUDA та OpenCL для подальшого покращення продуктивності.

OpenCV має потужну документацію та підтримку спільноти. Бібліотека добре задокументована з навчальними посібниками та зразками коду, щоб розробники могли почати якомога швидше. Активна спільнота підтримує форуми, репозиторії GitHub, щоб надавати допомогу та сприяти постійному розвитку.

Модульна структура OpenCV є ще однією ключовою особливістю, бібліотека розділена на кілька модулів, кожен з яких зосереджений на певних областях комп'ютерного зору. Ці модулі містять такі основні функції, як базові структури даних і обробка зображень, аналіз відео та машинне навчання.

Крім того, взаємодія OpenCV з іншими бібліотеками та фреймворками підвищує його універсальність. Наприклад, у Python його можна

використовувати разом із NumPy для числових операцій, що забезпечує потужну та гнучку обробку даних.

OpenCV досить широко використовується в задачах ідентифікації та класифікації зображень, де попередньо навчені моделі дозволяють розробникам створювати програми, здатні розпізнавати та класифікувати різні об'єкти в зображенні [18]. Його можливості ідентифікації облич і аналізу роблять його придатним для систем безпеки, програм соціальних мереж і аналітики клієнтів. Знову ж таки, OpenCV використовується для виявлення та відстеження об'єктів із його надійними методами аналізу відеопотоку в реальному часі; як наслідок, програма широко використовується у відеоспостереженні, автономних транспортних засобах та інтерактивних медіа [19].

Бібліотека також знаходить велике застосування в доповненій реальності; через неї доповнені об'єкти розміщуються на екрані поверх сцени. Ігри, освіта та роздрібна торгівля є основними сферами застосування, які вважають ці доповнення дуже ефективними. У медичній візуалізації ця бібліотека використовується для розробки інструментів для аналізу медичних зображень для подальшого вдосконалення процесу діагностики, планування лікування та дослідження. У сфері робототехніки OpenCV широко використовується для функціонування навігації та картографування разом із функціями взаємодії з навколишнім середовищем, щоб робот міг правильно сприймати та реагувати на навколишнє середовище.

Отже, OpenCV – це комплексна та універсальна бібліотека, яка рекомендувала себе як наріжний камінь у галузі комп'ютерного зору. Його широкі функціональні можливості, можливість роботи в реальному часі та підтримка багатьох платформ і мов роблять його безцінним інструментом для розробників і дослідників. Безперервна еволюція бібліотеки, керована жвавою спільнотою, гарантує, що вона залишається в авангардних технологічних досягненнях комп'ютерного зору та машинного навчання. Широкий спектр програм OpenCV, від ідентифікації зображення до

робототехніки, підкреслює його важливість і широке застосування як в академічній, так і в комерційній сферах.

2.2.2 Огляд основних функцій OpenCV для роботи з відеопотоком

Захоплення відео є основним етапом обробки відео, і це робиться за допомогою класу `cv2.VideoCapture`. Цей клас можна використовувати для захоплення відеопотоків з різних джерел: відео файлів або камер. Після захоплення відеопотоку до його кадрів можна отримати доступ і обробити їх за допомогою різних функцій обробки зображень у OpenCV [17, 20].

Запис оброблених кадрів назад у відеофайли реалізовано за допомогою класу `cv2.VideoWriter`, і цей процес можна розглядати як створення нового відеовмісту. Це найважливіше для редагування відео, оскільки цей процес передбачає збереження змін, внесених до оригінального відео.

Одним із фундаментальних завдань обробки відео є виявлення руху. Популярним алгоритмом реалізації виявлення руху є віднімання фону. OpenCV пропонує два таких підходи – `cv2.createBackgroundSubtractorMOG2` і `cv2.createBackgroundSubtractorKNN`, для сегментації рухомих об'єктів із фону. Ці інструменти знайшли багато застосувань у системах спостереження для виявлення зловмисників або в моніторингу.

OpenCV ефективний у виявленні та розпізнаванні об'єктів у відеопотоках. Каскадні класифікатори Хаара на основі функцій у `cv2.CascadeClassifier` реалізовані та забезпечують хороші результати для програм у реальному часі під час виявлення об'єктів, таких як обличчя та пішоходи.

2.3 Функціонал моделі YOLOv8 та побудова датасетів

2.3.1 Опис моделі YOLOv8 та її архітектури

YOLO (You Only Look Once) – це серія моделей виявлення об'єктів у реальному часі, відомих своєю швидкістю та точністю. Спочатку розроблені

Джозефом Редмоном і Алі Фархаді моделі YOLO пройшли кілька ітерацій, причому YOLOv8 є одним із останніх досягнень. YOLOv8 спирається на основи, закладені його попередниками, включаючи передові методи для підвищення продуктивності завдань виявлення об'єктів [21].

Розроблений для подолання обмежень попередніх версій YOLO, YOLOv8 забезпечує баланс між швидкістю та точністю [22]. Одним із значних удосконалень у YOLOv8 є його модульна та масштабована структура. Модель складається з трьох основних частин (рис. 2.1): хребта, шиї та голови. Хребет (магістраль) виконує завдання вилучення функцій із вхідного зображення за допомогою таких опцій, як CSPDarknet53 і EfficientDet. Шия, яка з'єднує хребет із головою, відіграє вирішальну роль у злитті функцій. Голова відповідає за прогнозування обмежувальних рамок, класів об'єктів і показників впевненості.

Важливим аспектом архітектури YOLOv8 є акцент на масштабуванні моделі. Він пропонує різні варіанти, такі як YOLOv8-tiny і YOLOv8x, кожен з яких відрізняється розміром і складністю обчислення. Це дозволяє користувачам вибрати модель, яка відповідає їхнім конкретним потребам, будь то середовища з обмеженими ресурсами чи високопродуктивні програми.

YOLOv8 також представляє вдосконалення в стратегіях навчання, включаючи такі методи, як оптимізація Rectified Adam (RAdam) і вибір між виявленням об'єктів на основі або без прив'язки. Ці вдосконалення призводять до швидшої конвергенції під час навчання та кращої продуктивності в задачах виявлення об'єктів.

Крім того, YOLOv8 має гнучку систему конфігурації, яка дозволяє користувачам налаштовувати різні параметри, включаючи розмір введення, блоки прив'язки та складність моделі. Ця адаптивність робить YOLOv8 придатним для широкого діапазону наборів даних і прикладних сценаріїв.

2.3.2 Огляд основних особливостей та переваг YOLOv8 для обробки відеопотоку

YOLOv8 розроблено для виявлення об'єктів у режимі реального часу, що важливо для таких програм, як спостереження, автономне водіння та аналіз відео в реальному часі, де своєчасна ідентифікація об'єктів є критичною. Як вже було визначено, архітектура є модульною та масштабованою, розділеною на три основні компоненти: хребет, шия та голова. Така конструкція забезпечує гнучкість у виборі та комбінуванні різних компонентів відповідно до конкретних потреб. Модель пропонує такі варіанти, як YOLOv8-tiny та YOLOv8x, що дозволяє користувачам балансувати між швидкістю та точністю відповідно до можливостей апаратного забезпечення та вимог програми.

Магістраль з такими опціями, як CSPDarknet53 і EfficientDet, оптимізована для вилучення багатих функцій із вхідних кадрів, гарантуючи, що навіть найдрібніші деталі будуть захоплені для підвищення точності виявлення. Компонент шиї призначений для ефективного об'єднання функцій, поєднання інформації з різних шарів для покращення виявлення об'єктів у кількох масштабах, що особливо корисно під час обробки відео, де розміри та вигляд об'єктів можуть суттєво відрізнятися.

Розширені функції YOLOv8 забезпечують значні переваги для обробки відео. Модель забезпечує високу точність виявлення об'єктів без шкоди для швидкості, що робить її ідеальною для додатків у режимі реального часу, де необхідні як швидка обробка, так і точне виявлення. Її масштабованість і модульність дозволяють використовувати його в широкому діапазоні програм обробки відео, від середовищ з обмеженими ресурсами до високопродуктивних систем.

Розширені стратегії навчання та гнучка конфігурація YOLOv8 дозволяють йому добре працювати з різними наборами даних. YOLOv8 можна налаштувати для досягнення оптимальної продуктивності, незалежно від того, чи йдеться про багатолюдні міські сцени чи рідкісні сільські

середовища. Завдяки потужним можливостям відстеження об'єктів YOLOv8 може підтримувати безперервність ідентичності об'єктів у кадрах, що має вирішальне значення для програм, які вимагають узгодженого моніторингу об'єктів.

Крім того, YOLOv8 оптимізовано для сучасних апаратних прискорювачів, таких як GPU і TPU, що гарантує, що він може використовувати найновіші обчислювальні ресурси для швидшої обробки. Ця адаптивність робить його перспективним вибором для додатків обробки відео, здатним задовольнити сучасні вимоги, що постійно розвиваються.

2.3.3 Опис методів побудови датасетів для тренування моделі YOLOv8

Створення високоякісних наборів даних має вирішальне значення для навчання ефективних моделей виявлення об'єктів, таких як YOLOv8. Добре підібраний набір даних гарантує, що модель може навчитися точно розпізнавати та виявляти об'єкти в різних сценаріях [23].

Першим кроком у створенні набору даних є збір відповідних даних. Для навчання YOLOv8 це передбачає збір зображень або кадрів, які містять об'єкти інтересу. Існує кілька підходів до збору даних:

- публічні набори даних: використання наявних загальнодоступних наборів даних може заощадити час і зусилля. Такі відомі набори даних, як COCO (Common Objects in Context), Pascal VOC і Open Images, надають широкий спектр позначених зображень, придатних для навчання моделей виявлення об'єктів. Ці набори даних часто використовуються як еталони в спільноті комп'ютерного зору та можуть стати міцною основою для навчання;

- збір спеціальних даних: для певних програм може знадобитися збір спеціальних даних. Це передбачає захоплення зображень або кадрів за допомогою камер або інших пристроїв для обробки зображень. Спеціальний збір даних дозволяє створити набір даних, який точно відповідає вимогам

програми, включаючи конкретні класи об'єктів, умови середовища та точки зору;

- генерація синтетичних даних: у деяких випадках генерація синтетичних даних може бути корисною. Цей метод використовує комп'ютерну графіку для створення реалістичних зображень об'єктів у різних умовах. Синтетичні дані можуть доповнювати дані реального світу, особливо коли важко охопити певні сценарії або варіації об'єктів.

Після того, як дані зібрані, наступним кроком є анотація, яка передбачає позначення об'єктів на зображеннях. Точні та послідовні анотації мають вирішальне значення для ефективного навчання моделі. Анотація може виконуватися вручну або напівавтоматично:

- для анотації вручну використовують такі інструменти, як LabelImg, RectLabel або VGG Image Annotator, щоб намалювати рамки навколо об'єктів і призначити мітки класу. Цей метод забезпечує високу якість анотацій, але може бути тривалим і трудомістким;

- напівавтоматична анотація зазвичай реалізована за допомогою таких інструментів, як Amazon SageMaker Ground Truth або Labelbox. Ці інструменти використовують попередньо навчені моделі, щоб запропонувати обмежувальні прямокутники та мітки, які можна потім перевірити та налаштувати за потреби. Цей підхід може пришвидшити процес анотації, зберігаючи при цьому точність.

Щоб підвищити різноманітність і надійність набору даних, застосовуються методи доповнення даних. Розширення збільшує різноманітність навчальних прикладів без збору нових даних, допомагаючи моделі краще узагальнювати. Загальні методи збільшення включають:

- геометричні перетворення: застосування до зображень таких перетворень, як обертання, масштабування, перевертання та обрізання. Ці зміни допомагають моделі стати інваріантною до різних орієнтацій і розмірів об'єктів;

- трансформації колірного простору: налаштування властивостей кольору зображень, таких як яскравість, контраст, насиченість і відтінок. Ці зміни дозволяють моделі працювати зі змінами умов освітлення;

- введення шуму: додавання випадкового шуму або розмивання частин зображення для імітації різних умов зображення. Ця техніка допомагає моделі стати більш стійкою до різних якостей зображення;

- CutMix і MixUp: ці вдосконалені методи доповнення поєднують частини різних зображень або змішують два зображення, щоб створити нові навчальні приклади. Це може покращити здатність моделі виявляти об'єкти в складних і зашумованих сценах.

Організація та підготовка набору даних є останнім кроком перед навчанням моделі YOLOv8. Правильна організація забезпечує ефективне завантаження та обробку під час навчання:

- упорядкування набору даних у структуровану ієрархію каталогу. Загальним підходом є наявність окремих папок для наборів навчання, перевірки та тестування з підкаталогами для кожного класу об'єктів;

- сумісність формату анотації із YOLOv8. Формат YOLO зазвичай включає текстові файли з кожним зображенням, де кожен рядок відповідає об'єкту та містить мітку класу та нормалізовані координати обмежувальної рамки;

- поділ набору даних на набори навчання, перевірки та тестування. Загальний розподіл: 70 % для навчання, 20 % для перевірки та 10 % для тестування. Цей поділ гарантує, що модель оцінюється на основі небачених даних, допомагаючи оцінити її можливості узагальнення.

Створення високоякісного набору даних для навчання YOLOv8 передбачає ретельне планування та виконання збору даних, анотації, доповнення та організації. Правильна організація та підготовка набору даних додатково забезпечує ефективне навчання та оцінку, що зрештою призводить до більш точної та надійної моделі виявлення об'єктів.

2.4 Висновки до розділу

У даному розділі кваліфікаційної роботи було проведено детальний аналіз та порівняння існуючих технологій ідентифікації номерних знаків автомобілів з точки зору точності, швидкодії та інших критичних параметрів. На основі цього аналізу було обрано мову програмування та відповідні бібліотеки для реалізації системи.

На основі аналізу різних мов програмування було обрано Python. Цей вибір обґрунтований популярністю і наявністю спеціалізованих бібліотек для реалізації машинного навчання та комп'ютерного зору.

Серед багатьох доступних бібліотек для Python була обрана OpenCV, яка повністю підходить для реалізації системи ідентифікації номерних знаків. Було описано архітектуру моделі YOLOv8 та розглянуто її особливості і переваги в обробці відео. У дослідженні також наведено загальну характеристику методів створення набору даних для тренування моделі YOLOv8. Широкий функціонал OpenCV та гнучкість YOLOv8 допоможе досягти продуктивність та ефективність розробленої системи, що є ключовими вимогами для практичного застосування у реальних умовах

3 РОЗРОБЛЕННЯ ПРОГРАМНОГО МОДУЛЯ ІДЕНТИФІКАЦІЇ НОМЕРНИХ ЗНАКІВ АВТОТРАНСПОРТНИХ ЗАСОБІВ

3.1 Вибір відповідної моделі виявлення об'єктів

Вибір відповідної моделі виявлення об'єктів є критичним кроком у розробці надійної системи ідентифікації автотранспортного засобу. Система повинна мати можливість ідентифікувати та відстежувати присутність автотранспортних засобів у різних налаштуваннях, забезпечуючи високу точність та ефективність. З доступних варіантів моделі було обрано попередньо підготовлену модель YOLOv8n на основі набору даних COCO. Пояснення цього вибору, враховуючи особливі вимоги до ідентифікації об'єкту, і переваги, які пропонує модель YOLOv8n, детально описано в цьому підрозділі.

Модель YOLOv8n є варіантом сімейства YOLO (You Only Look Once), розробленого з урахуванням принципу підтримки компромісу між швидкістю та точністю швидкості. Це робить його ідеальним кандидатом для застосунків у режимі реального часу, як, наприклад, застосунок для ідентифікації автотранспортного засобу.

Буква «n» у YOLOv8n означає «нано», що значить, що цей конкретний варіант моделі високо оптимізований для швидкості та ефективності. У системах ідентифікації автотранспортних засобів у режимі реального часу обробка відеопотоків має відбуватися надзвичайно швидко, щоб забезпечити своєчасну ідентифікацію та відстеження. Легка архітектура YOLOv8n робить його зручним для використання на пристроях з обмеженими обчислювальними ресурсами, як-от вбудовані системи та мобільні пристрої.

Незважаючи на компактні розміри, YOLOv8n гарантує високий рівень точності. Модель побудована таким чином, що вона може належним чином виявляти автотранспортні засоби навіть у складних сценаріях, таких як різні

умови освітлення, різні кути та оклюзії. Навчання набору даних COCO ще більше підвищує його надійність, оскільки під час навчання він бачив широкий спектр зразків зображень автотранспортних засобів.

Набір даних COCO різноманітний і містить понад 200 000 зображень із мітками 80 різних категорій об'єктів. Ця різноманітність гарантує, що модель YOLOv8n може добре узагальнюватися для різних середовищ і умов, які виникають у реальних програмах. Те, що модель може точно підібрати автомобіль, буде ефективно використовуватися в міських, приміських і сільських умовах.

Надійні можливості виявлення YOLOv8n виходять за межі простого визначення присутності автотранспортних засобів. Це гарантує, що модель може успішно знаходити автотранспортні засоби за допомогою обмежувальних рамок (рис. 3.1), які надають точну інформацію про те, де їх знайти. Це має ключове значення для таких застосувань, як системи допомоги при паркуванні та інтелектуальне керування дорожнім рухом, у яких потрібна точна інформація для подальшого процесу.

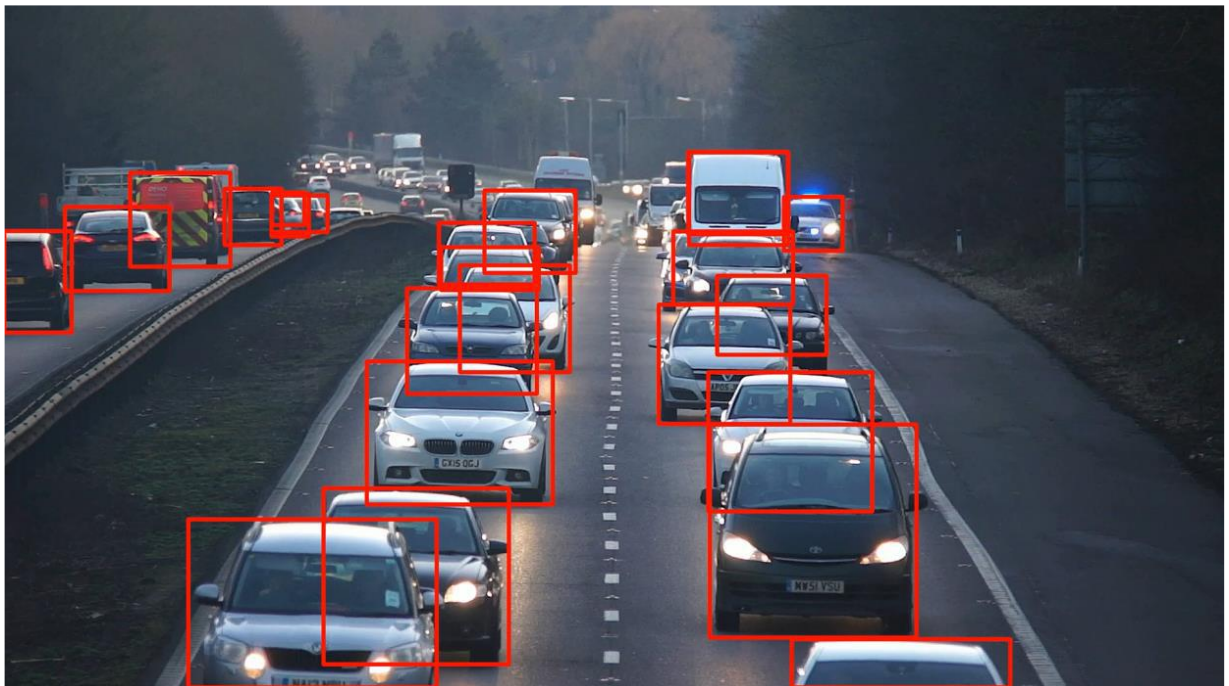


Рисунок 3.1 – Ідентифікації автотранспортних засобів моделлю YOLOv8n

3.2 Вибір алгоритму відстеження об'єктів

Для побудови цілісної системи ідентифікації номерних знаків необхідно більше, ніж просто виявлення автотранспортного засобу. Важливо мати можливість отримувати ці результати, зберігаючи об'єкти узгоджено в усіх кадрах у відеопотоці. Належне відстеження об'єктів гарантує точне відстеження кожного виявленого автотранспортного засобу під час його руху в полі зору камери. Серед багатьох розглянутих алгоритмів було обрано SORT (Simple Online and Realtime Tracking).

SORT передбачає майбутні положення відстежуваних об'єктів за допомогою фільтра Калмана, тобто навіть у разі оклюзії або коли об'єкти тимчасово залишають поле зору, він може підтримувати точне відстеження.

Угорський алгоритм, по суті, призначає виявлені об'єкти існуючим шляхам, метод, який підтримує оптимальність призначення пар і гарантує, що витрати зводяться до мінімуму, таким чином уникаючи неправильного призначення шляху.

Як видно зі схеми (рис. 3.2), у кожному кадрі автотранспортних засобів спочатку виявляються за допомогою YOLOv8n, і виводяться відповідні граничні рамки. Ці виявлені рамки передаються в SORT, який ініціалізує нові шляхи для раніше невиявлених автотранспортних засобів або оновлює положення існуючих шляхів автотранспортного засобу. Цей послідовний процес допомагає системі добре працювати в динамічних середовищах.

Можливість передбачення фільтра Калмана всередині SORT допомагає зберігати шляхи транспортних засобів, навіть якщо вони тимчасово закриті іншими об'єктами. Крім того, механізм асоціації даних алгоритму сортує транспортні засоби, які знову з'являються після оклюзії, щоб забезпечити постійне та надійне відстеження.

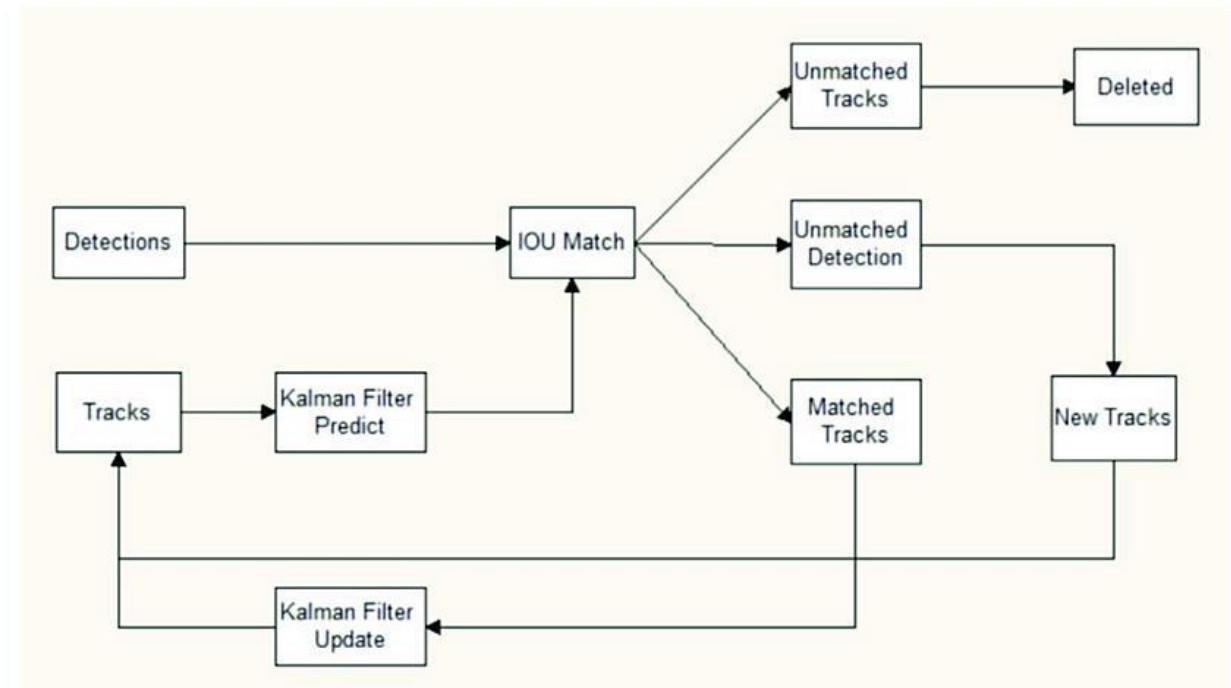


Рисунок 3.2 – Процедура відстеження об'єктів алгоритмом SORT

Вибір алгоритму SORT у системі ідентифікації автотранспортного засобу пропонує баланс між простотою, ефективністю та надійністю. Його продуктивність у режимі реального часу, простота впровадження та сумісність із моделлю виявлення YOLOv8n роблять його ідеальним вибором для відстеження автотранспортних засобів у відеопотоці. Завдяки інтеграції SORT з YOLOv8n система може досягти точного та надійного виявлення та відстеження автотранспортних засобів. Цей вибір гарантує, що система ідентифікації автотранспортного засобу працює ефективно, зберігаючи високу продуктивність навіть у динамічних і складних середовищах.

3.3 Реалізація моделі ідентифікації об'єктів

3.3.1 Огляд спеціального набору даних

Для досягнення високої точності та ефективності визначення об'єктів було обрано модель YOLOv8n та спеціальний набір даних для її тренування. Спеціальний датасет під назвою `resized640_aug3x-ACCURATE` був обраний з онлайн джерела «Roboflow». Зважаючи на його опис на сайті, цей набір

даних має кілька переваг і може бути хорошим вибором для розробки тренованої моделі ідентифікації об'єктів.

Описаний набір даних структуровано за допомогою детального складу та методології попередньої обробки, спрямованої на покращення процесу навчання для виявлення об'єктів за допомогою моделі YOLO. Набір даних розділений на три частини: навчальний набір, набір перевірки та тестовий набір. Навчальний набір складається з 87% набору даних, що становить 21 174 зображення. Це основна підмножина, яка використовується для навчання моделі. Набір перевірки, який використовується для точного налаштування параметрів моделі та запобігання переобладнанню, складається з 8 % набору даних, що означає 2048 зображень. Нарешті, тестовий набір, який має вирішальне значення для остаточної оцінки продуктивності моделі, становить 4 % набору даних із 1020 зображень.

Етапи попередньої обробки ретельно застосовуються для забезпечення однорідності та готовності зображень для моделі (рис. 3.3).



Рисунок 3.3 – Зразок модифікованих зображень тренувального набору

Одним із етапів попередньої обробки є автоматична орієнтація, яка регулює орієнтацію зображень на основі їхніх метаданих, забезпечуючи їхню правильну орієнтацію. Цей крок допомагає підтримувати узгодженість у наборі даних. Крім того, розмір усіх зображень змінюється шляхом їх розтягування до фіксованого розміру 640×640 пікселів, стандартизації

вхідного розміру для моделі та полегшення пакетної обробки під час навчання.

Для подальшого вдосконалення набору даних застосовано низку методів доповнення даних. Кожен навчальний приклад доповнюється втричі за допомогою різних перетворень для збільшення варіативності набору даних. Застосовується горизонтальне гортання, створюючи дзеркальну версію зображення, що фактично подвоює різноманітність даних. Кадрування передбачає збільшення зображення з мінімальним масштабуванням 0 % і максимальним масштабуванням 15 %, імітуючи різні масштаби об'єктів і точки зору. Обертання застосовується в діапазоні від -10° до $+10^\circ$, що додає варіативності та робить модель більш стійкою до змін орієнтації об'єкта. Зсув дещо спотворює зображення як у горизонтальному, так і у вертикальному напрямках у межах $\pm 2^\circ$, допомагаючи моделі справлятися зі змінами перспективи.

Перетворення градацій сірого застосовується до 10 % зображень, покращуючи здатність моделі виявляти об'єкти незалежно від кольору. Також передбачено коригування відтінку, насиченості, яскравості та експозиції, кожне в межах певного діапазону, щоб імітувати різні умови освітлення та налаштування камери. Ці коригування разом підвищують надійність моделі. Гаусове розмиття з радіусом до 0,5 пікселя моделює сценарії незначного розмиття, допомагаючи моделі працювати з різною якістю зображення. Нарешті, збільшення вирізу вводить оклюзії, застосовуючи п'ять рамок вирізів на зображення, кожна з яких займає 2 % площі зображення. Це змушує модель навчитися виявляти об'єкти, навіть якщо частини відсутні, тим самим покращуючи її здатність узагальнювати різні сценарії.

Включаючи ці етапи попередньої обробки та доповнення, набір даних має на меті надати різноманітний і вичерпний набір навчальних прикладів. Цей підхід покращує здатність моделі до узагальнення та надійність, роблячи її більш ефективною в задачах виявлення реальних об'єктів.

3.3.2 Тренування моделі ідентифікації об'єктів

Результати тренування моделі можна побачити у Додатку А, у таблиці А.1. Навчання моделі YOLOv8n протягом 20 епох із спеціальним набором даних включає кілька ключових кроків:

- датасет необхідно організувати в певну структуру каталогів. Як показано на схемі (рис. 3.4), набір даних повинен мати окремі папки для зображень і міток для навчання, перевірки та тестування. Кожен файл мітки має відповідати назві відповідного файлу зображення та мати розширення «.txt» з даними анотації у форматі YOLO.

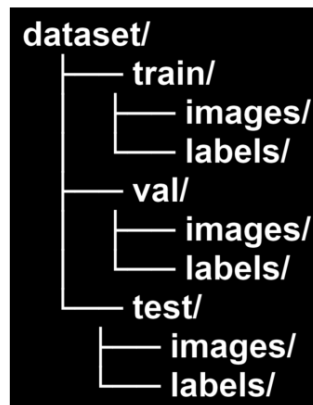


Рисунок 3.4 – Специфіка структури директорії для навчання

- необхідно створити файл «data.yaml», щоб указати шляхи до набору даних та інші деталі конфігурації. Як видно з рисунка (рис. 3.5), цей файл YAML має включати шляхи до каталогу зображень навчання, перевірки та тестування, кількість класів «nc» і список імен класів;

```

train: ../train/images
val: ../valid/images
test: ../test/images

nc: 1
names: ['License_Plate']
  
```

Рисунок 3.5 – Файл конфігурації для навчання моделі

Аналізуючи результати тренування (рис. 3.6) можливо зробити висновок, що модель набула майже оптимальних значень під час 20-ї епохи. Продовжувати навчання не є бажаним, бо з'явиться ризик перенавчання, що призведе до зменшення точності моделі.

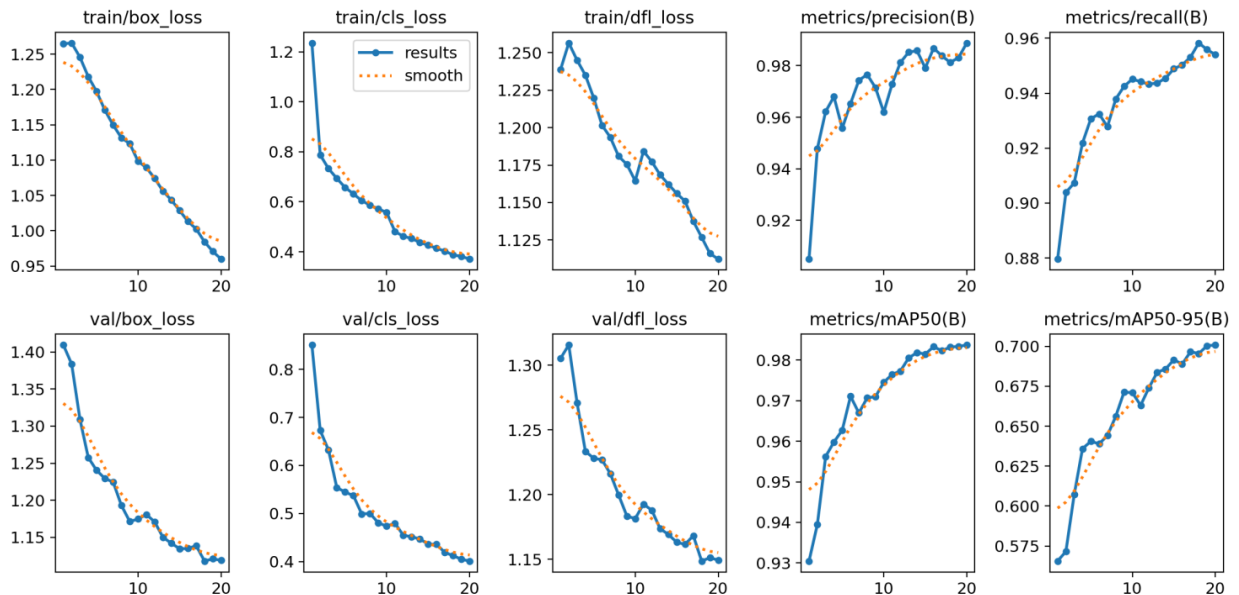


Рисунок 3.6 – Графічне представлення результатів тренування моделі

3.3.3 Оцінка результатів моделі на валідаційних даних

Оцінка ефективності навченої моделі є критично важливим кроком у процесі розробки, що дає змогу зрозуміти її точність, надійність і можливість узагальнення. Ретельна оцінка для системи виявлення номерних знаків, що використовує модель YOLOv8n, гарантує, що модель добре працює не лише на навчальних даних, але й у сценаріях реального світу.

Було проведено порівняння з даними перевірки, щоб оцінити наскільки добре модель узагальнює нові, невідомі дані (рис 3.7 – 3.9). Модель, яка добре працює на навчальному наборі, але погано на перевірочному наборі, ймовірно, перенавчена, тобто вона вивчила навчальні дані надто конкретно й може не працювати добре в реальних програмах.

Моніторинг показників перевірки під час навчання дозволяє достроково припинити навчання, коли навчання припиняється, якщо

продуктивність моделі на наборі перевірки перестає покращуватися. Це допомагає запобігти перенавчанню та гарантує, що модель підтримує хороше узагальнення.



Рисунок 3.7 – Валідаційні та передбачені мітки даних на val_batch0



Рисунок 3.8 – Валідаційні та передбачені мітки даних на val_batch1



Рисунок 3.9 – Валідаційні та передбачені мітки даних на val_batch2

Оцінка моделі за даними перевірки допомагає аналізувати зсувність (помилки через надто спрощені припущення в моделі) і дисперсію (помилки через надмірну складність і чутливість до навчальних даних). Баланс між зсувом і дисперсією є вирішальним для надійної роботи моделі.

З графічного порівняння валідаційних даних і виводу навченої моделі можна зробити висновок, що вона повністю впоралася з поставленою задачею. Також слід зауважити, що тренувана модель фіксує навіть ті знаки, які не знаходяться в фокусі наданих зображень, але вже з меншим коефіцієнтом впевненості.

3.4 Огляд проблеми ідентифікації об'єктів

Системи ідентифікації об'єктів (LPR) відіграють вирішальну роль у різних програмах, таких як контроль за дотриманням правил дорожнього руху, автоматизований збір плати, керування паркуванням і відстеження автотранспортних засобів. Незважаючи на значні досягнення в області комп'ютерного зору та машинного навчання, системи LPR все ще стикаються з кількома проблемами, зокрема з точним ідентифікаціям символів на

номерних знаках. Найбільш серйозною проблемою в системах LPR є візуальна схожість між певними символами. Наприклад, «O» і «0», «I» і «1», «J» і «3», «A» і «4», «G» і «6», «S» і «5» можна легко сплутати один з одним. Ця проблема посилюється через відмінності шрифтів, умов освітлення та якості зображення.

Щоб вирішити конкретну проблему візуально схожих символів, було реалізовано спеціальний етап постобробки. Після початкової ідентифікації номерного знаку за допомогою EasyOCR розпізнані символи були зіставлені з їх найімовірнішими аналогами на основі попередньо визначених правил. Наприклад:

- “O” було зіставлено з “0”;
- “I” було зіставлено з “1”;
- “J” було зіставлено з “3”;
- “A” було зіставлено з “4”;
- “G” було зіставлено з “6”;
- “S” було зіставлено з “5” і навпаки.

Були реалізовані перевірки на основі очікуваного формату номерних знаків. Відомо, що номерний знак має певний шаблон, показаний на (рис. 3.10), будь-яке відхилення від цього шаблону позначалося, а правила відображення символів застосовувалися для виправлення потенційних помилок.

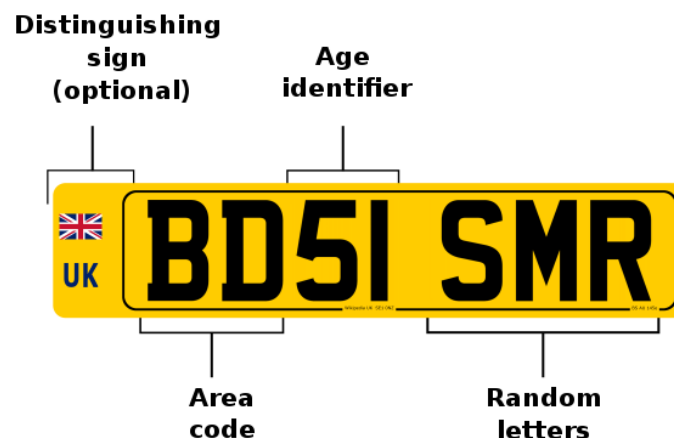


Рисунок 3.10 – Очікуваний формат номерного знаку

Отже, реалізоване рішення передбачало інтеграцію EasyOCR із спеціальним відображенням символів і процесом перевірки. Порядок роботи був таким:

- були зроблені зображення номерних знаків та попередньо оброблені для покращення читабельності;
- попередньо оброблені зображення були завантажені в EasyOCR, який спровокував початкове ідентифікації символів разом із їх оцінками впевненості;
- розпізнані символи було оброблено за допомогою спеціальних правил відображення, щоб виправити будь-які ймовірні помилки ідентифікації. Контекстна перевірка підтвердила, що виправлена послідовність відповідає очікуваним форматам номерних знаків;
- остаточним результатом стала виправлена послідовність символів, що представляють номерний знак.

3.5 Важливість інтерполяції даних для належної візуалізації результатів

Інтерполяція даних є фундаментальною технікою аналізу та візуалізації даних, особливо коли мова йде про дискретні точки даних, які потрібно представити безперервно. Інтерполяція допомагає оцінити проміжні значення між відомими точками даних, таким чином створюючи більш плавну та повну візуалізацію.

У контексті проекту була використана лінійна інтерполяція для заповнення невідомих координат обмежувальних рамок під час відстеження автотранспортного засобу, що рухається.

Лінійна інтерполяція – це метод оцінки невідомих значень, які знаходяться між двома відомими значеннями. Для цілей цього проекту його використовували для інтерполяції координат обмежувальних рамок для кадрів, де не було визначено точне положення автотранспортного засобу.

Кілька причин підкреслюють доречність лінійної інтерполяції в цьому сценарії:

- лінійна інтерполяція є простою для реалізації та ефективною з точки зору обчислень. Враховуючи необхідність обробки відеокадрів у реальному часі, простота лінійної інтерполяції гарантує, що обчислювальні витрати залишаються мінімальними. Це має вирішальне значення для таких програм, як відстеження автотранспортного засобу в режимі реального часу, де важливо підтримувати високу частоту кадрів;

- коли автомобіль рухається, його траєкторію між двома послідовними виявленими положеннями часто можна апроксимувати як лінійну, особливо протягом коротких інтервалів часу. Лінійна інтерполяція ефективно зберігає природну траєкторію руху автотранспортного засобу, забезпечуючи прямолінійну оцінку між відомими положеннями обмежувальної рамки. Це гарантує, що інтерпольовані рамки точно відповідають фактичній траєкторії автотранспортного засобу, не додаючи нереалістичних кривих або відхилень;

- лінійна інтерполяція особливо ефективна, коли проміжки між виявленими кадрами невеликі. У проекті система відстеження іноді пропускає кілька кадрів через оклюзії або тимчасову втрату виявлення. Для таких коротких проміжків лінійна інтерполяція забезпечує надійний метод оцінки положення автотранспортного засобу без значних помилок, зберігаючи безперервність координат обмежувальної рамки.

На практиці лінійна інтерполяція була реалізована в алгоритмі відстеження для обробки випадків, коли модель виявлення не могла ідентифікувати автомобіль в одному або кількох послідовних кадрах. Це гарантувало, що координати обмежувальної рамки залишаються точними та послідовними протягом усього відеоряду.

3.6 Перевірка та аналіз роботи програмного модуля

Програмний модуль виявлення номерних знаків розроблено як консольний додаток, що забезпечує просте розгортання та інтеграцію у великі системи. Модуль працює, обробляючи відео, виявляючи атранспортні засоби та їхні номерні знаки, а потім виводячи відео з підсвічуванням розпізнаних номерних знаків. Алгоритм роботи модуля наведено на рис. 3.11.

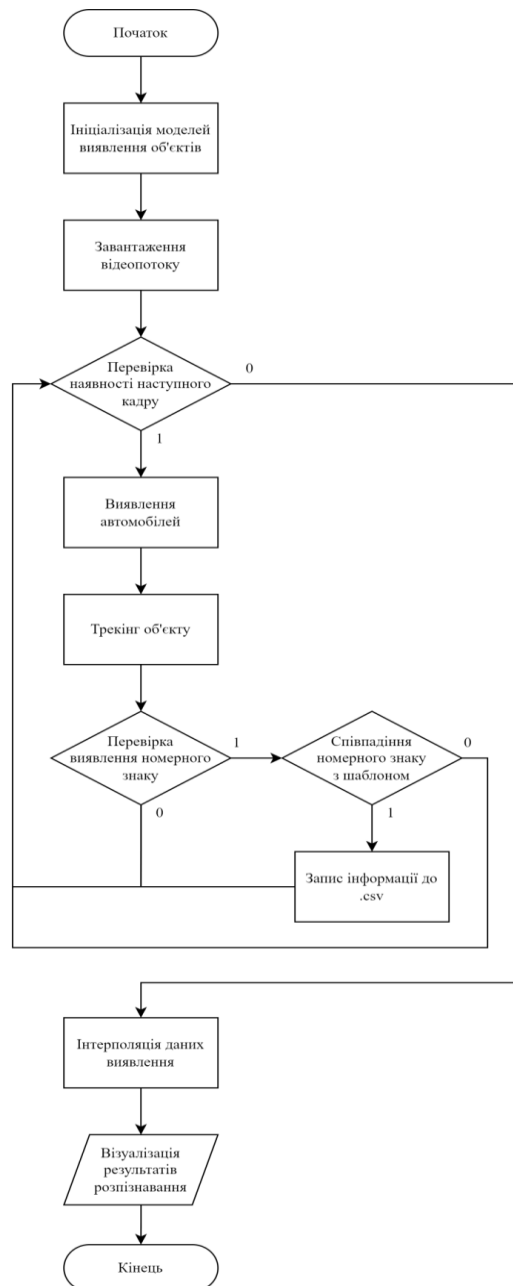


Рисунок 3.11 – Алгоритм роботи програмного модуля

Алгоритм роботи програмного модуля:

- модуль приймає відеофайл як вхідний та відкриває його;
- кожен кадр відео обробляється для виявлення автотранспортних засобів та їхніх номерних знаків. Модель YOLOv8n, попередньо навчена на спеціальному наборі даних, використовується для виявлення номерних знаків у кожному кадрі;
- ідентифікації номерних знаків відбувається за використання бібліотеки EasyOCR. Методи постобробки застосовуються для вирішення неоднозначності символів;
- оброблені кадри з виділеними номерними знаками компілюються у вихідний відеофайл. Цей відеофайл забезпечує візуальне представлення продуктивності модуля.

Для оцінки ефективності модуля ідентифікації номерних знаків було проведено тестування з використанням відео транспортного потоку. Результати роботи модуля можна побачити на рис. 3.12 – 3.14, метою було оцінити точність і ефективність модуля в реальному сценарії.



Рисунок 3.12 – Фрагмент результуючого відео



Рисунок 3.13 – Фрагмент результуючого відео
з неправильним ідентифікацієм



Рисунок 3.14 – Фрагмент результуючого відео

Модуль досяг високої точності виявлення, правильно ідентифікуючи більшість номерних знаків на відео. Деякі проблеми були відмічені у

випадках сильної оклюзії або поганого освітлення, що іноді призводило до пропусків виявлення.

Точність ідентифікації також була високою, більшість символів на номерних знаках ідентифікувалися правильно. Спеціальні правила постобробки ефективно вирішували загальні неоднозначності. Однак траплялися незначні помилки у випадках забруднених номерних знаків.

3.7 Висновки до розділу

У цьому розділі було проведено огляд моделі YOLOv8, попередньо навченої на наборі даних COCO. Вона була обрана завдяки балансу швидкості та точності, необхідного для того, щоб зробити модель практичною в реальних умовах.

Алгоритм SORT буде безперервно та стабільно відстежувати цілі навіть із повним або частковим закриттям і втратою виявлення в певних кадрах, таким чином адаптуючись до високодинамічних сцен і придатний для більшості застосувань у моніторингу руху.

Модель YOLOv8n було навчено на спеціальному наборі даних. Методи збільшення даних були включені, щоб зробити модель універсальною. Процес навчання був зосереджений на точному налаштуванні моделі для забезпечення високої точності виявлення та локалізації номерних знаків, забезпечуючи надійну систему виявлення.

Результат валідації показав, що модель добре узагальнює нову інформацію, запобігаючи випадкам перенавчання. Ця ретельна перевірка підтвердила, що модель ефективна для вирішення цього завдання та надійна у застосуванні в сценаріях реального часу.

При наявності розрізнявальних неоднозначностей символів, які досить візуально схожі, в модуль ідентифікації номерних знаків за допомогою бібліотеки EasyOCR було реалізовано контекстна перевірка.

Система тестувалася на відеофайлі з автотранспортними засобами, демонструючи високу точність у реальних умовах. Таким чином, розроблено надійну систему, придатну для застосування в управлінні дорожнім рухом і автоматизованому відстеженні автотранспортних засобів.

4 ЗАХОДИ І РОЗРАХУНКИ ДЛЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕЧНИХ УМОВ ПРАЦІ

4.1 Аналіз умов праці на робочому місці

На робочому місці оператора ПК згідно виникають небезпечні та шкідливі фактори: підвищений рівень шуму, несприятливі мікрокліматичні умови, недостатній рівень освітленості, шкідливі речовини, підвищений рівень електромагнітних випромінювань радіочастот, висока напруга електричної мережі, статична електрика та інші. Робота з ПК супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці оператора ПК є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів.

Організація робочого місця. Приміщення, в якому знаходиться робоче місце оператора ПК, загальною площею 48 м², і висотою стелі 3,5 м. У приміщенні знаходиться 6 робочих місць з ПК. Кожне робоче місце обладнане робочим столом, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші.

4.2 Промислова безпека на робочому місці

Живлення ПК здійснюється від трифазної чотирьох електричної мережі змінного струму з глухо-заземленою нейтраллю і напругою 220 В, частотою 50 Гц. Згідно НПАОП 40.1-1.21-98 приміщення можна віднести до

категорії без підвищеної небезпеки, так як в приміщенні відсутні чинники, які викликають підвищену або особливу небезпеку.

Для створення безпечних умов праці необхідно провести ряд організаційних і технічних заходів. Згідно НПАОП 40.1-1.32-01 для запобігання ураження людини електричним струмом в приміщенні застосовується система занулення.

4.3 Виробнича санітарія у приміщенні

Робота оператора ПК за енерговитратами відноситься до категорії легких робіт. В таблиці 4.1 наведені оптимальні параметри мікроклімату в приміщеннях, де виконуються роботи операторського типу [24].

Таблиця 4.1 – Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	22 – 24 °С; 40 – 60 %; до 0,1 м/с
Теплий	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	23 – 25 °С; 40 – 60 %; 0,1 – 0,2 м/с

Виміряні за допомогою приладів температура та вологість у лабораторії відповідають вказаним у таблиці для теплого періоду року. Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціонування повітря, або забезпечити подачу свіжого повітря системами вентиляції.

Лабораторія, де виконується розробка конструкції модуля, має наступні характеристики:

– площа приміщення 48 м² (8×6 м);

- висота – 3,5 м;
- кількість робочих місць – 6 шт.;
- обладнання – стіл з ПК і периферією – 6 шт.

Приміщення, відповідно до ДНАОП 0.00-1.31-99, має забезпечувати 6 м² площі та 20 м³ обсягу на одне окреме робоче місце з ПК [24]. Площа приміщення 48 м² та об'єм 168 м³, на кожне робоче місце приходиться 8 м² площі і об'єм 28 м³, тобто вимога виконана.

Приміщення з ПК повинні мати природне і штучне освітлення відповідно до ДБН В.25-28-2006 «Природне і штучне освітлення». Природне світло повинно проникати через бічні світлові прорізи, зорієнтовані, як правило, на північ або північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5 %.

Рівень загального штучного освітлення приміщення можна перевірити за допомогою методу питомої потужності, викладеної в [24].

Розрахункова формула методу:

$$W = \frac{W_{\Sigma}}{S}, \quad (4.1)$$

де W – питома потужність, Вт/м²;

S – площа приміщення, м²;

W_{Σ} – загальна потужність освітлювальної установки Вт, яка розраховується за формулою:

$$W_{\Sigma} = W_{ce} \cdot n_{ce}, \quad (4.2)$$

де W_{ce} – потужність одного світильника, Вт;

n_{ce} – кількість світильників в приміщенні.

$$W_{\Sigma} = 100 \cdot 4 = 400 \text{ Вт}, \quad (4.3)$$

$$W = \frac{400}{48} = 8,33 \text{ Вт/м}^2. \quad (4.4)$$

Питомої потужності 8,33 Вт/м² по таблиці Б.3 із [24] відповідає освітленість в 250 лк при мінімальній допустимій освітленості 300 лк.

Отже, для створення сприятливих зорових умов в лабораторії необхідно збільшити кількість світильників або замінити лампи в світильниках на більш потужні.

4.4 Пожежна безпека виробничого приміщення

Пожежна безпека – стан об'єкта, при яким виключається можливість пожежі, а у випадку його виникнення запобігає вплив на людей небезпечних факторів пожежі й забезпечується захист матеріальних цінностей.

Пожежна безпека забезпечується системою запобігання пожежі й системою пожежного захисту. У всіх службових приміщеннях обов'язково повинен бути «План евакуації людей при пожежі», що регламентує дії персоналу у випадку виникнення вогнища загоряння, що й указує місця розташування пожежної техніки.

Горючими компонентами у виробничому приміщенні є: перегородки, двері, підлоги, ізоляція кабелів тощо.

Протипожежний захист – це комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобігання пожежі, обмеження його поширення, а також на створення умов для успішного гасіння пожежі.

Джерелами запалювання у виробничому приміщенні можуть бути електронні схеми від ПК, прилади, застосовувані для технічного обслуговування, пристрою електроживлення, кондиціонування повітря, де в

результаті різних порушень утворюються перегріті елементи, електричні іскри й дуги, здатні викликати загоряння горючих матеріалів.

У сучасних ПК дуже висока щільність розміщення елементів електронних схем. У безпосередній близькості друг від друга розташовуються сполучні проведення, кабелі. При протіканні по них електричного струму виділяється значна кількість теплоти. При цьому можливо оплавлення ізоляції. Для відводу надлишкової теплоти від ПК служать системи вентиляції й кондиціонування повітря. При постійній дії ці системи являють собою додаткову пожежну небезпеку.

Енергопостачання виробничого приміщення здійснюється за допомогою трансформаторної станції та за допомогою двигун-генераторних агрегатів. На трансформаторних підстанціях особливу небезпеку представляють трансформатори які мають масляне охолодження. У зв'язку із цим перевагу слід віддавати сухим трансформаторам.

ВИСНОВКИ

Кваліфікаційна робота присвячена дослідженню та власній реалізації програмного модулю ідентифікації номерних знаків автотransпортних засобів, яка базується на сучасних методах комп'ютерного зору. У ході дослідження було проведено огляд існуючих підходів до ідентифікації номерних знаків, розроблено алгоритми для обробки зображень і ідентифікації символів, а також проведено експериментальну перевірку ефективності запропонованих рішень.

Перший розділ роботи заклав теоретичну основу для розробки програмного модулю ідентифікації номерних знаків автотransпортних засобів. Було визначено основні компоненти сучасної системи ALPR, такі як виявлення номерного знака, сегментація та ідентифікації символів. Аналіз основних компонентів системи та порівняння методів машинного навчання та архітектур нейронних мереж дали всебічне розуміння існуючих підходів та викликів у розпізнаванні та ідентифікації номерних знаків автотransпортних засобів. Це знання стало базою для подальшої розробки та реалізації ефективної системи LPR.

У другому розділі було проведено детальний аналіз існуючих технологій розпізнавання та ідентифікації номерних знаків автотransпортних засобів з точки зору точності, швидкодії та інших критичних параметрів. В результаті було обрано мову програмування Python та бібліотеку OpenCV для реалізації програмного модуля. Архітектура моделі YOLOv8 та її особливості і переваги в роботі з відео були розглянуті, що дозволило обґрунтовано обрати цю модель для використання в проєкті. Опис методів створення набору даних для тренування моделі допоміг забезпечити продуктивність та ефективність програмного модуля.

У третьому розділі було проведено огляд моделі YOLOv8, попередньо навченої на наборі даних COCO, та обрано алгоритм SORT для стабільного

та безперервного відстежування автотранспортних засобів. Модель YOLOv8n було навчено на спеціальному наборі даних, із застосуванням методів збільшення даних для покращення універсальності моделі. Процес навчання був зосереджений збільшенні точності виявлення та локалізації номерних знаків автотранспортних засобів. Валідація показала, що модель добре узагальнює нову інформацію та запобігає перенавчанню. Для вирішення проблеми ідентифікації візуально схожих символів було реалізовано контекстну перевірку за допомогою бібліотеки EasyOCR.

У четвертому розділі були визначені небезпечні та шкідливі виробничі чинники. Таким чином, були розроблені заходи і технічні засоби щодо забезпечення безпеки праці працюючого персоналу. Застосування цих заходів дасть змогу максимально знизити ймовірність отримання травм під час роботи, а також поліпшити умови роботи персоналу лабораторії, де виконувалася кваліфікаційна робота.

Програмний модуль був протестован на відеофайлі з автотранспортними засобами, продемонструвавши високу точність у реальних умовах. Вихідне відео, з анотованими номерними знаками, підтвердило ефективність і надійність модуля. Цей тест підтвердив, що програмний модуль здатен працювати в реальних умовах, забезпечуючи розпізнавання та точну ідентифікацію номерних знаків автотранспортних засобів.

Загалом, кваліфікаційна робота продемонструвала ефективність та надійність розробленого програмного модулю розпізнавання та ідентифікації номерних знаків автотранспортних засобів. Вибір моделей та алгоритмів, їх тренування на спеціально підготовлених наборах даних, а також ретельна перевірка результатів підтвердили готовність системи до практичного застосування. Модуль може бути інтегрований у більші транспортні системи для забезпечення автоматизації процесів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015 Інформація та документація «Звіти у сфері науки і техніки». Структура та правила оформлювання. / В. Земцева; Ю. Поліщук, канд. фіз.-мат. наук; Р. Санченко, канд. техн. наук; Л. Шрамко; А. Ямчук (науковий керівник) ДП «УкрНДНЦ» від 22 червня 2015р. № 61 з 2017- 07-01.

2. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, О.І. Филипенко, О.В. Токарева, С.П. Новоселов, О.В Сичова. – Харків: ХНУРЕ, 2023. – 64 с.

3. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології»: Навчальний посібник / І. Ш. Невлюдов, В. А. Андрусевич, О. В. Токарева, С. П. Новоселов, О. В. Сичова. – Харків : Видавництво Іванченка І. С., 2022. – 151 с.

4. Технічні засоби автоматизації: Підручник / І. Ш. Невлюдов, А. О. Андрусевич, О. І. Филипенко, Н. П. Демська, С. П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 366 с.

5. What is license plate recognition?. Survisiongroup [Електронний ресурс] / Режим доступу: <https://survisiongroup.com/post-what-is-license-plate-recognition> – 15.04.2025 р. – Загл. з екрану.

6. License Plate Recognition From Still Images and Video Sequences: A Survey / C. N. E. Anagnostopoulos et al. IEEE Transactions on Intelligent Transportation Systems. Vol. 9, no. 3. 2008. – PP. 377–391.
7. Kumari P. Automated License Plate Recognition (ALPR) Model: A Complete Guide. Labellerr [Электронный ресурс] / Режим доступа: <https://www.labellerr.com/blog/automated-license-plate-recognition-in-automotive-industry-tutorial/> – 17.04.2025 г. – Загл. з экрана.
8. Patel M. The Complete Guide to Image Preprocessing Techniques in Python. Medium [Электронный ресурс] / Режим доступа: <https://medium.com/@maahip1304/the-complete-guide-to-image-preprocessing-techniques-in-python-dca30804550c> – 18.04.2025 г. – Загл. з экрана.
9. Suguna R., Devi M. S. Image Preprocessing and Its Applications in Computer Vision. Journal of Computational and Theoretical Nanoscience. Vol. 15, no. 11, 2018. – PP. 3381–3388.
10. k-NN classifier for image classification – PyImageSearch. PyImageSearch [Электронный ресурс] / Режим доступа: <https://pyimagesearch.com/2016/08/08/k-nn-classifier-for-image-classification/> – 18.04.2025 г. – Загл. з экрана.
11. Machine Learning OCR – Intelligent Text Detection 2.0. Klippa [Электронный ресурс] / Режим доступа: <https://www.klippa.com/en/information/ocr-machine-learning/> – 20.04.2025 г. – Загл. з экрана.
12. Mark Lutz. Learning Python: Powerful Object-Oriented Programming, 6th Edition / Publisher(s): O'Reilly Media, Inc., 2018. – 736 p.
13. David Beazley. Python Cookbook: Recipes for Mastering Python 3 / Publisher(s): O'Reilly Media, Inc., 2020. – 521 p.
14. Bjarne Stroustrup. The C++ Programming Language, 4rd Edition / Publisher(s): Addison-Wesley Professional, 2013. – 1376 p.
15. John Paxton, Adam D. Scott & Shelley Powers. JavaScript Cookbook: Programming the Web. 3rd Ed. / Publisher(s): O'Reilly Media, Inc., 2021. – 650 p.

16. Brahmhatt S. Practical OpenCV. Berkeley, CA: Apress, 2013 [Електронний ресурс] / Режим доступу: <https://doi.org/10.1007/978-1-4302-6080-6> – 24.04.2025 р. – Загл. з екрану.
17. OpenCV modules. OpenCV [Електронний ресурс] / Режим доступу: <https://docs.opencv.org/4.x/> – 25.04.2025 р. – Загл. з екрану.
18. Bose S. Traffic Detection Using OpenCV. SSRN Electronic Journal. 2020 [Електронний ресурс] / Режим доступу: <https://doi.org/10.2139/ssrn.3604892> – 25.04.2025 р. – Загл. з екрану.
19. OpenCV – бібліотека комп'ютерного зору і машинного навчання з відкритим вихідним кодом [Електронний ресурс] / Режим доступу: <https://opencv.org/> – 25.04.2025 р. – Загл. з екрану.
20. OpenCV Tutorial: A Guide to Learn OpenCV in Python. Great Learning Blog: Free Resources what Matters to shape your Career! [Електронний ресурс] / Режим доступу: <https://www.mygreatlearning.com/opencv-tutorial-in-python/> – 26.04.2025 р. – Загл. з екрану.
21. Spodarets D. A Guide to the YOLO Family of Computer Vision Models. Data Phoenix [Електронний ресурс] / Режим доступу: <https://dataphoenix.info/a-guide-to-the-yolo-family-of-computer-vision-models/> – 27.04.2025 р. – Загл. з екрану.
22. Ultralytics. YOLOv8. Home – Ultralytics YOLO Docs [Електронний ресурс] / Режим доступу: <https://docs.ultralytics.com/models/yolov8/#supported-tasks-and-modes> – 28.04.2025 р. – Загл. з екрану.
23. Ultralytics. Datasets Overview. Home – Ultralytics YOLO Docs [Електронний ресурс] / Режим доступу: <https://docs.ultralytics.com/datasets/#new-ultralytics-explorer> – 29.04.2025 р. – Загл. з екрану.
24. Комплекс навчально-методичного забезпечення навчальної дисципліни «Організація керування умовами праці» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т.Є. Стиценко, Г.В. Пронюк, Н.М. Сердюк. – Харків, 2017. – 108 с.