

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод визначення вільних місць на відкритих паркуваннях
нейромережевими засобами

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

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

До захисту допускаю:
зав. кафедри КН, д.т.н., професор  Олександр БАРМАК
Підпис Ім'я, ПРІЗВИЩЕ

17 06 2025 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук


(підпис)

д.т.н., професор Олександр БАРМАК

« 10 » 02 2025 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами»

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

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

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


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


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

Мета кваліфікаційної роботи бакалавра – підвищення ефективності визначення вільних паркувальних місць на відкритих стоянках за допомогою нейромережесвих методів. Завдання на кваліфікаційну роботу бакалавра: провести аналіз сучасних нейромережесвих підходів до оброблення зображень для виявлення вільних місць на паркуваннях; спроектувати метод визначення наявності вільних місць на відкритих паркуваннях з використанням згорткової нейронної мережі; розробити архітектуру для реалізації запропонованого методу; реалізувати програмне забезпечення відповідно до спроектованої архітектури; провести тестування застосунку на відкритих наборах зображень паркувальних майданчиків.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

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

Виконавець: студент групи КН-21-1  Микола КУЧЕР
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами»

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

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

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
56	21	1	43	2

Мета кваліфікаційної роботи – підвищення точності виявлення вільних паркувальних місць на відкритих майданчиках шляхом розробки методу, заснованого на нейромережових технологіях. Метод ґрунтується на поєднанні виявлення транспортних засобів за допомогою нейронної мережі YOLO з аналізом задалегідь визначених координат паркувальних місць, що дозволяє автоматично встановлювати їхній стан – зайняте або вільне.

Розроблено програмний застосунок на мові Python із використанням бібліотек Tkinter (для створення графічного інтерфейсу), Ultralytics YOLO (для виявлення об'єктів) та OpenCV (для обробки зображень). Застосунок дозволяє обробляти зображення з камер або фото майданчиків і визначати доступні паркомісця в автоматичному режимі. Рішення спрямоване на покращення ефективності використання паркувального простору в міських умовах.

Ключові слова: вільні паркувальні місця, нейромережі, YOLO, автоматичне виявлення, OpenCV.

Виконавець: студент групи КН-21-1
Група виконавця


Підпис

Микола КУЧЕР
Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень.....	3
Вступ.....	4
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій .5	5
1.1 Аналіз інформаційних моделей	5
1.2 Огляд нейромережових технологій для аналізу зображень.....	7
1.3 Аналіз існуючих рішень для визначення вільних місць на відкритих паркуваннях	10
1.4 Мета та завдання кваліфікаційної роботи	14
Розділ 2 Проектування методу визначення вільних місць на відкритих паркуваннях нейромережовими засобами	15
2.1 Загальна концепція та схема методу визначення вільних паркувальних місць	15
2.2 Архітектура нейромережової моделі для обробки зображень паркінгу	17
2.3 Підготовка даних для донавчання моделі	20
2.4 Виявлення паркувальних місць у зображеннях	23
2.5 Класифікація вільних паркувальних місць	26
2.6 Метрики оцінювання ефективності моделі.....	29
2.7 Висновки до розділу 2	33
Розділ 3 Особливості реалізації та результати тестування методу визначення вільних місць на відкритих паркуваннях	34
3.1 Особливості розробки програмного застосунку для імплементації запропонованого методу	34
3.1.1 Засоби розробки програмного застосунку	34
3.1.2 Програмна реалізація та інтерфейс користувача	37
3.2 Результати тестування методу визначення вільних паркувальних місць.....	42
3.2.1 Результати донавчання моделі YOLO на навчальній вибірці.....	42
3.2.2 Результати тестування моделі YOLO на тестовій вибірці.....	47
3.3 Висновки до розділу 3	51
Загальні висновки	52
Перелік посилань	53
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
YOLO	You Only Look Once
API	Application Programming Interface
CNN	Convolutional Neural Network
OpenCV	Open Source Computer Vision Library
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
SSD	Single Shot Detector
GPS	Global Positioning System
CRM	Customer Relationship Management
R-ELAN	Residual Enhanced Lightweight Attention Network
FPN	Feature Pyramid Network
CSP	Cross Stage Partial
ХНУ	Хмельницький національний університет
PAN	Path Aggregation Network
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
macOS	Macintosh Operating System
YAML	YAML Ain't Markup Language
CPU	Central Processing Unit
GPU	Graphics Processing Unit
IoU	Intersection over Union

Вступ

Стрімке зростання кількості автомобілів у містах призводить до перевантаження транспортної інфраструктури, зокрема паркувальних майданчиків. Відсутність автоматизованих систем моніторингу часто ускладнює процес паркування, що, у свою чергу, спричиняє затори, підвищені витрати пального та негативний вплив на екологію. Перспективним напрямом у цьому контексті є застосування методів комп'ютерного зору та глибокого навчання, зокрема нейронних мереж, які забезпечують високу точність аналізу візуальної інформації.

Об'єкт дослідження – процес автоматичного визначення вільних паркувальних місць на відкритих паркувальних майданчиках за допомогою методів комп'ютерного зору.

Предмет дослідження – нейромережеві моделі та алгоритми комп'ютерного зору, що використовуються для аналізу зображень паркувальних майданчиків з метою виявлення зайнятих та вільних місць.

Мета кваліфікаційної роботи бакалавра – підвищення точності визначення вільних паркувальних місць на відкритих паркуваннях шляхом проєктування методу аналізу стану паркувальних місць із використанням нейромережевих технологій та створення програмного застосунку на його основі.

Завдання кваліфікаційної роботи бакалавра: провести аналіз нейромережевих моделей та існуючих підходів для визначення вільних місць на відкритих паркуваннях; розробити метод визначення вільних місць на відкритих паркуваннях з використанням нейромережевої моделі; підготувати набір даних, що містить різноманітні зображення відкритих паркуваннях для донавчання моделі; провести валідацію запропонованого методу за відомими статистичними показниками.

Інтелектуальні системи моніторингу паркування оптимізують міський простір, скорочуючи час пошуку місця та знижуючи навантаження на дороги й довкілля.

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

1.1 Аналіз інформаційних моделей

У сучасних містах проблема ефективного використання паркувального простору стає дедалі актуальнішою. Зростання кількості автомобілів призводить до перевантаження паркувальних майданчиків, особливо у центральних районах. Відсутність оперативної інформації про наявність вільних місць спричиняє додаткові витрати часу для водіїв та збільшує трафік у пошуках паркування.

Відкриті паркувальні майданчики, які не мають фізичних бар'єрів або шлагбаумів, є поширеними в українських містах. Наприклад, у Львові активно облаштовують нові паркувальні майданчики, зокрема на вулицях Городоцькій та Вітовського, що свідчить про постійне зростання потреби в організованому паркуванні [1].



Рисунок 1.1 – Паркувальний майданчик у Львові [2]

Традиційні методи контролю паркування, такі як ручний облік або використання шлагбаумів, не забезпечують достатньої ефективності та не дозволяють в режимі реального часу відстежувати наявність вільних місць. Це створює необхідність впровадження інноваційних технологій для автоматичного визначення зайнятості паркувальних місць.

Одним із перспективних напрямків є використання нейромережових технологій для аналізу зображень з камер спостереження. Такі системи можуть автоматично визначати, чи зайняте, паркувальне місце на основі обробки відеопотоку.

Іншим підходом є використання технологій Edge Computing, де обробка даних відбувається безпосередньо на пристрої, розташованому на паркувальному майданчику [3]. Це дозволяє зменшити затримки в передачі даних та забезпечити більш оперативне оновлення інформації про наявність вільних місць.

Загалом, впровадження нейромережових технологій та Edge Computing у сфері паркування дозволяє:

- зменшити час пошуку вільного паркувального місця для водіїв;
- знизити рівень трафіку та викидів CO₂ у містах;
- підвищити ефективність використання паркувального простору;
- забезпечити оперативну інформацію для міських служб та водіїв [4].

Також варто зазначити, що компанія Parkinto пропонує комерційне рішення для виявлення вільних паркувальних місць з використанням камер та нейромережових алгоритмів. Система дозволяє в режимі реального часу отримувати інформацію про зайнятість паркувальних місць та надає API для інтеграції з іншими сервісами [5].

Розглянемо основні сутності та процеси предметної області, а також визначимо ключові аспекти, які можна автоматизувати з використанням сучасних методів машинного навчання та штучного інтелекту.

Паркувальні місця – фізичні одиниці на відкритому паркуванні, які можуть бути вільними або зайнятими. Вони є основними одиницями вимірювання та обробки в системах [6].

Камери – спостереження пристрої, які здійснюють постійний моніторинг стану паркувальних місць, забезпечуючи збір зображень для подальшої обробки нейронними мережами [7].

Моделі нейронних мереж – обчислювальні структури, які аналізують Вхідні зображення, визначають наявність транспортних засобів на паркувальних місцях та приймають рішення про їхній статус (вільне або зайняте) [8].

Збір даних – камери спостереження фіксують зображення паркувальних місць на реальному паркуванні [9].

Попередня обробка зображень – зображення проходять обробку для підготовки до подальшого аналізу (наприклад, корекція освітлення, видалення шумів) [10].

Аналіз зображень – нейронна мережа проводить аналіз зображень, визначаючи, чи зайняте паркувальне місце транспортним засобом [11].

Постановка результатів – на основі аналізу система надає відповідь про статус кожного паркувального місця (вільне або зайняте), ця інформація може бути доступна водіям у реальному часі через мобільні додатки або інформаційні табло на паркуванні [12].

Таким чином, автоматизація процесу визначення вільних місць на відкритих паркувальних майданчиках за допомогою нейромережових технологій є актуальним та перспективним напрямком розвитку міської інфраструктури.

1.2 Огляд нейромережових технологій для аналізу зображень

Завдання автоматичного визначення вільних паркувальних місць є актуальним у контексті розвитку "розумної" міської інфраструктури. Сучасні інформаційні технології дозволяють ефективно вирішувати це завдання завдяки засобам штучного інтелекту, комп'ютерного зору та гнучким інструментам

розробки. Реалізація програмної системи здійснюється з використанням бібліотек для обробки зображень і відеопотоків (таких як OpenCV, NumPy, TensorFlow або PyTorch), а також моделей глибокого навчання, зокрема згорткових нейронних мереж (CNN), що продемонстрували високу ефективність у задачах комп'ютерного зору [13].

Центральним елементом рішення є застосування згорткових нейронних мереж, зокрема моделі YOLO (You Only Look Once), яка здатна одночасно виявляти кілька об'єктів на зображенні з високою точністю та швидкістю. У межах проекту використовуватиметься YOLOv12 — одна з найновіших і найбільш оптимізованих версій моделі, доступна через бібліотеку ultralytics. Завдяки цій моделі можливо здійснювати розпізнавання транспортних засобів у реальному часі на зображеннях з камер відеоспостереження [14].

Приклад використання YOLO на рисунку 1.2.

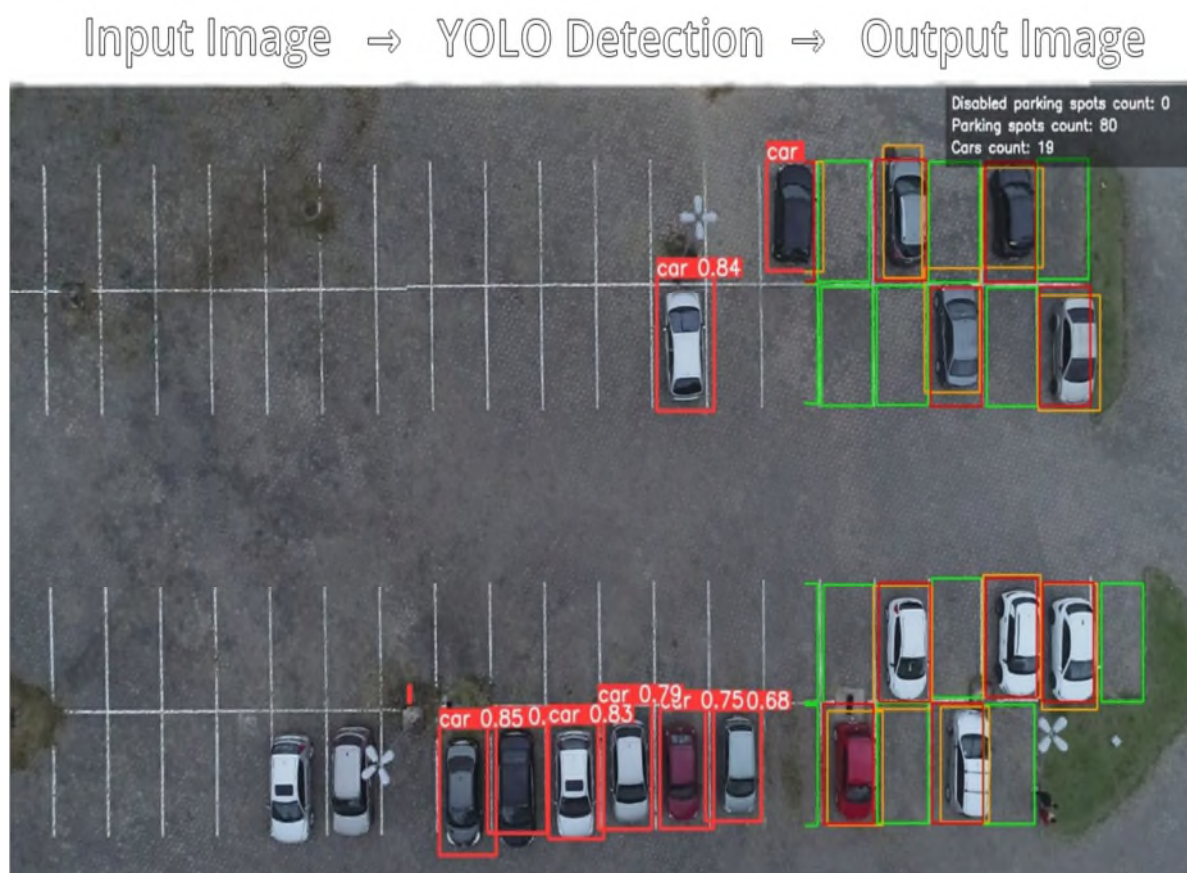


Рисунок 1.2 – Розпізнавання об'єктів за допомогою YOLO[15]

Це забезпечує основу для оцінки статусу кожного паркувального місця як "зайняте" або "вільне".

Для захоплення відео з камер та попередньої обробки кадрів використовуються бібліотеки OpenCV. OpenCV (Open Source Computer Vision Library) – це потужна бібліотека для комп'ютерного зору, яка широко застосовується для обробки зображень і відео в реальному часі. Вона дозволяє здійснювати попередню обробку зображень, таку як корекція освітлення, фільтрація шумів, нормалізація перспективи та інші важливі операції. OpenCV також підтримує захоплення відео з камер, що є важливим для проекту, де потрібно відстежувати вільні і зайняті паркувальні місця на реальних відео [16].

Алгоритми включатимуть також попередню фільтрацію зображень – для зменшення шумів, покращення освітлення або нормалізації перспективи. Це критично важливо для підвищення точності розпізнавання автомобілів у складних умовах, таких як дощ, нічна зйомка або нестандартне розташування камер [17].

Розпізнавання вільних та зайнятих паркувальних місць за допомогою нейронних мереж. Використання нейромереж для автоматичного визначення статусу паркувальних місць на основі аналізу зображень і відео з паркувальних майданчиків. Використання технологій комп'ютерного зору для реального часу. Створення систем для моніторингу паркувальних місць, які здійснюють розпізнавання вільних місць у реальному часі на відеопотоках з камер спостереження [18].

Інтеграція з мобільними додатками для водіїв. Розробка мобільних застосунків, що отримують дані з системи моніторингу і показують водіям інформацію про вільні паркувальні місця на карті або в режимі реального часу. Оптимізація паркувальних ресурсів за допомогою аналізу даних. Застосування алгоритмів машинного навчання для аналізу даних про заповненість паркувальних майданчиків, що дозволяє прогнозувати рівень заповнення та оптимізувати розподіл паркувальних місць [19].

YOLOv12 забезпечить швидке й точне виявлення автомобілів. Інтеграція OpenCV як інструменту для обробки зображень дозволяє значно покращити якість отриманих даних та адаптувати систему до різних умов зйомки, що в свою чергу підвищує ефективність і точність роботи моделі.

1.3 Аналіз існуючих рішень для визначення вільних місць на відкритих паркуваннях

На сьогодні автоматизація паркувального контролю є важливим напрямом у розвитку "розумного міста", і з кожним роком з'являється все більше рішень, спрямованих на зменшення часу пошуку паркувального місця, оптимізацію трафіку та підвищення зручності для водіїв. Аналіз існуючих інформаційних систем та наукових розробок дозволяє визначити, які технологічні підходи є найбільш ефективними і які з них доцільно використовувати для реалізації власної системи.

Умовно всі системи можна класифікувати на кілька типів:

1. Системи з апаратною детекцією (GPS, сенсори, магнітні детектори);
2. Системи на основі відеоаналітики з класичними методами обробки зображень;
3. Системи з використанням штучного інтелекту та глибокого навчання (зокрема YOLO, SSD, Faster R-CNN) [20].

Найбільш перспективними сьогодні вважаються рішення на основі глибокого навчання, що дозволяють проводити автоматичний аналіз відео з камер спостереження без додаткового апаратного забезпечення. Тепер перейдемо до зразків, які схожі на проект.

Smart Parking Detection System with YOLOv8 and OpenCV – це програмне рішення, розроблене ViduraSC як відкрите програмне забезпечення, що працює локально без підключення до хмари. Система реалізована з використанням сучасної версії моделі YOLOv8 для виявлення транспортних засобів у реальному

часі з відеопотоку. Для захоплення та обробки кадрів використовується бібліотека OpenCV, а для створення графічного інтерфейсу – PyQt5 [21].

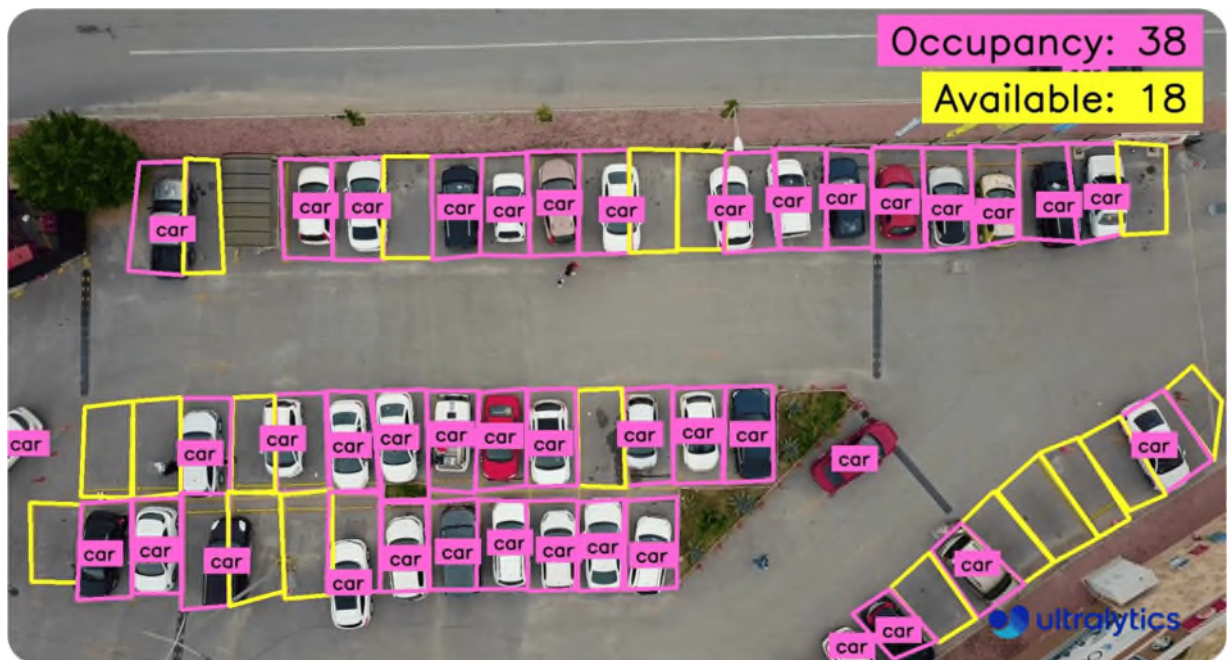


Рисунок 1.3 – Розпізнавання вільних місць [22]

Функціональні можливості:

- автоматичне розпізнавання автомобілів;
- підрахунок зайнятих і вільних місць;
- кольорове маркування паркувальних місць;
- підтримка відео в реальному часі.

Переваги:

- повністю автономна робота без підключення до хмари;
- проста інтеграція з існуючими камерами відеоспостереження;
- висока точність при денному освітленні.

Недоліки:

- вразливість до змін умов освітлення (вночі, під дощем);
- низька адаптивність до нетипових кутів огляду камери;
- наявність базової графіки інтерфейсу — інтерфейс більше функціональний, ніж зручний для користувача.

Smart Parking System with Pixel-wise ROI and YOLOv8–YOLOv11 — це наукове дослідження, розроблене Gustavo P. C. P. da Luz та іншими дослідниками, що поєднує точкову (pixel-wise) сегментацію паркувальних місць із сучасними моделями YOLOv8–YOLOv11. Такий підхід дозволяє досягати надзвичайно високої точності — майже 100% — на спеціалізованих наборах даних. Сегментація паркувальних місць дозволяє точніше визначати їхній стан, а навчання на локалізованих зображеннях підвищує ефективність системи в конкретному середовищі. Публікація доступна на платформі arXiv. На рисунку 1.4 продемонстровано роботу програми [23].



Рисунок 1.4 – Робота програми[24]

Функціональні можливості:

- поєднання сегментації та об'єктного розпізнавання;
- тренування на локалізованих наборах даних;
- підтримка моделей від YOLOv8 до YOLOv11;
- робота в різноманітних умовах та з різними типами камер.

Переваги:

- надзвичайно висока точність (99.68%);
- універсальність під різні архітектури камер;

- адаптація до різних погодних умов.

Недоліки:

- високі обчислювальні вимоги;
- необхідність маркування кожного паркувального місця вручну для тренування;
- низька швидкодія на слабкому обладнанні.

Smart Parking System with License Plate Recognition – це академічна публікація, створена Висоцьким В. та Яворським Н., яка орієнтована не лише на визначення зайнятості паркувальних місць, а й на розпізнавання номерних знаків транспортних засобів. У системі використовується YOLO для детекції автомобілів, а також OCR-алгоритм для зчитування номерів і збереження їх у базі даних. Такий підхід забезпечує ефективний контроль в'їзду та виїзду, що робить систему ідеальною для закритих паркінгів. Також підтримується інтеграція з шлагбаумами та CRM-системами, а збереження даних може здійснюватися у хмарі [25].

Функціональні можливості:

- виявлення автомобілів за допомогою YOLO;
- розпізнавання номерних знаків за допомогою OCR;
- запис даних у базу;
- можливість інтеграції з шлагбаумами та системами доступу.

Переваги:

- підвищення безпеки;
- можливість інтеграції з системами шлагбаумів та CRM;
- підтримка хмарного збереження.

Недоліки:

- сильна залежність від якості відео;
- можливі помилки розпізнавання номерів при поганій погоді чи забрудненні авто.

Аналіз існуючих програмних рішень та досліджень свідчить про високу ефективність моделей YOLO (особливо останніх версій) у поєднанні з OpenCV

для реалізації систем контролю паркувальних місць. Однак жодне з рішень не є універсальним – більшість з них мають суттєві обмеження в умовах низької освітленості, нестандартного розміщення камер або великого навантаження на систему [26].

Найбільш актуальним напрямом, відповідно до проведеного аналізу, є розробка системи візуального моніторингу паркувальних місць з мінімальними вимогами до інфраструктури, але високою точністю та швидкістю.

1.4 Мета та завдання кваліфікаційної роботи

Метою кваліфікаційної роботи бакалавра є підвищення точності визначення вільних місць на відкритих паркуваннях із використанням згорткових нейронних мереж YOLOv12, що виконує функції виявлення, класифікації та візуалізації паркувальних місць на основі відеопотоку з камер спостереження.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- провести аналіз нейромережових моделей та існуючих підходів для визначення вільних місць на відкритих паркуваннях;
- розробити метод визначення вільних місць на відкритих паркуваннях з використанням нейромережової моделі;
- підготувати набір даних, що містить різноманітні зображення відкритих паркуваннях для донавчання моделі;
- провести валідацію запропонованого методу за відомими статистичними показниками.

Розробка такої системи дозволить значно підвищити ефективність управління відкритими паркувальними майданчиками, зменшити кількість пошукового трафіку, а також покращити комфорт та інформованість водіїв у межах міської інфраструктури.

Розділ 2 Проктування методу визначення вільних місць на відкритих паркуваннях нейромережевими засобами

2.1 Загальна концепція та схема методу визначення вільних паркувальних місць

Метод визначення вільних паркувальних місць на відкритих паркінгах базується на використанні згорткової нейромережі YOLOv12 для швидкої та точної детекції вільних паркувальних місць на зображеннях із камер відеоспостереження. Концепція передбачає аналіз зображення паркінгу, виявлення автомобілів, порівняння їхніх координат із заздалегідь визначеними зонами паркувальних місць і класифікацію місць як "вільних" або "зайнятих".

На рисунку 2.1 зображено концептуальну схему методу, яка включає такі основні етапи: вхідні дані(зображення із камери відеоспостереження), попередня обробка зображення, детекція автомобілів, аналіз зайнятості паркувальних місць та виведення вихідні дані (візуалізація вільних/зайнятих місць).

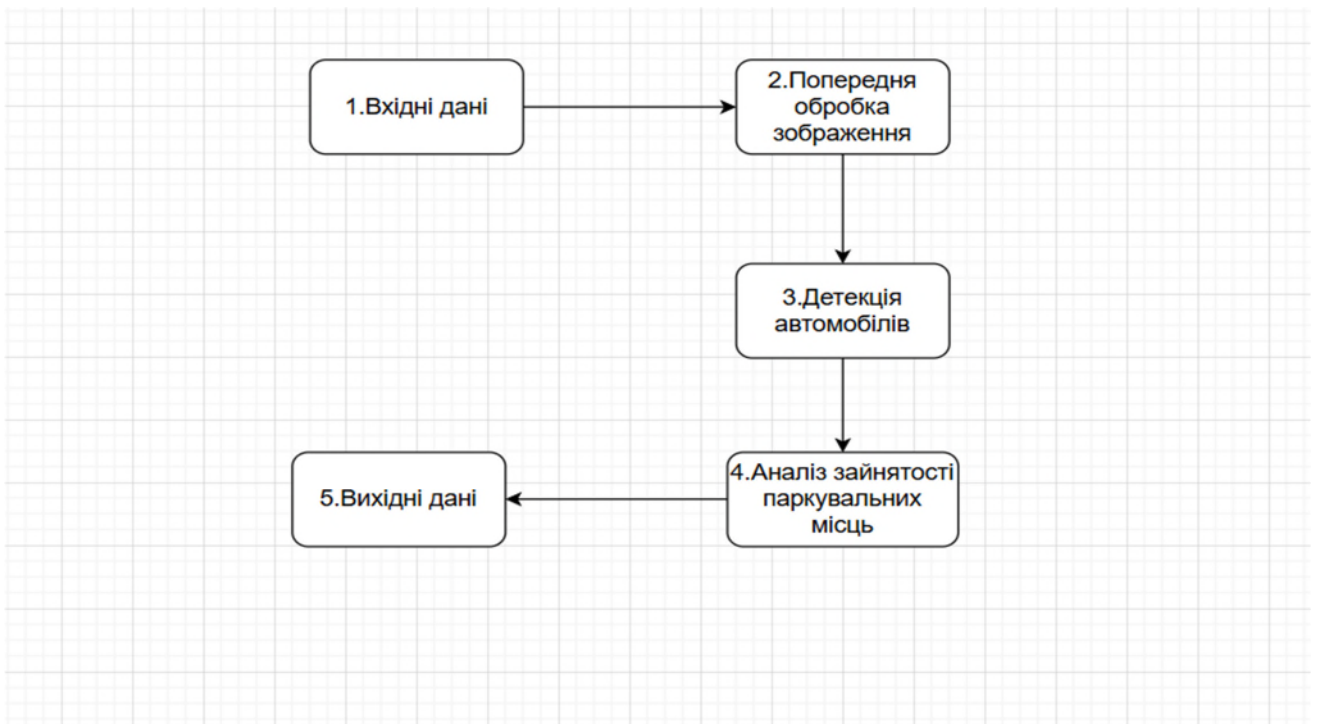


Рисунок 2.1 – Загальна схема методу визначення вільних місць на відкритих паркуваннях нейромережевими засобами

Розпізнавання вільних паркувальних місць виконується за допомогою глибокої нейронної мережі YOLO. Метод реалізується послідовно у п'ять етапів, що відповідають елементам схеми:

Крок 1. Вхідні дані

На першому етапі отримується зображення з камери відеоспостереження. Камера повинна бути встановлена таким чином, щоб охоплювати всю зону паркування – зверху або під кутом, який дозволяє чітко бачити кожне місце. Знімок повинен мати достатню роздільну здатність і бути без сильних спотворень, щоб система могла точно аналізувати ситуацію на парковці.

Крок 2. Попередня обробка зображення

Перед подачею в нейронну мережу, зображення проходить попередню обробку.

Крок 3. Детекція автомобілів

Оброблене зображення передається на вхід моделі YOLO (You Only Look Once). YOLO виконує детекцію автомобілів на зображенні та формує обмежувальні прямокутники (bounding boxes) навколо кожного об'єкта. Для кожної рамки вказується клас об'єкта (наприклад, автомобіль) і рівень впевненості в розпізнаванні. Це дозволяє точно і швидко виявити всі транспортні засоби на парковці.

Крок 4. Аналіз зайнятості паркувальних місць

Далі відбувається порівняння координат виявлених автомобілів із заздалегідь визначеними зонами паркувальних місць (прямокутниками або полігонами, які відповідають кожному реальному місцю для паркування). Якщо рамка автомобіля перекриває область місця – воно вважається зайнятим. Якщо перекриття відсутнє – місце вважається вільним. Таким чином, метод автоматично визначає статус кожного паркувального місця.

Крок 5. Вихідні дані

На завершальному етапі результати візуалізуються: вільні місця позначаються зеленим кольором, зайняті – червоним. Це дозволяє швидко орієнтуватися у поточній ситуації на парковці. Результати можуть передаватися в мобільні додатки, веб-інтерфейси або системи інформування водіїв.

Завдяки поетапній обробці зображення – від захоплення кадру до візуалізації результатів – метод дозволяє точно визначати зайнятість кожного паркувального місця.

2.2 Архітектура нейромережевої моделі для обробки зображень паркінгу

У сучасних системах комп'ютерного зору завдання виявлення транспортних засобів на паркувальних майданчиках за допомогою статичних зображень набуває особливої актуальності. Це обумовлено необхідністю автоматизованого моніторингу зайнятості паркувальних місць, що сприяє оптимізації використання паркувального простору та підвищенню ефективності міської інфраструктури. Серед існуючих підходів, модель YOLOv12 (You Only Look Once, версія 12) вирізняється високою точністю та швидкістю обробки, що робить її придатною для застосування в реальному часі.

Архітектура моделі YOLOv12, зображена на схемі на рисунку 2.2, складається з трьох головних частин: мережі екстракції ознак (Backbone), блоку агрегації просторових ознак (Neck) та голови прогнозування (Head). Вхідне зображення, як правило, має стандартні розміри (наприклад, 640×640 пікселів) і подається до згорткових шарів початкового рівня [27]. У Backbone'i використовується модифікована структура R-ELAN, що поєднує класичні згортки з залишковими з'єднаннями, а також часткову агрегацію ознак (CSP), яка дозволяє зменшити обчислювальну складність, не втрачаючи при цьому якість виділених ознак. У результаті формується багатоканальна карта ознак, де зберігається інформація про просторову структуру зображення, колірні переходи та контури

об'єктів. Зі схеми видно, що Backbone генерує ознаки на трьох різних масштабах, які передаються далі у Neck.

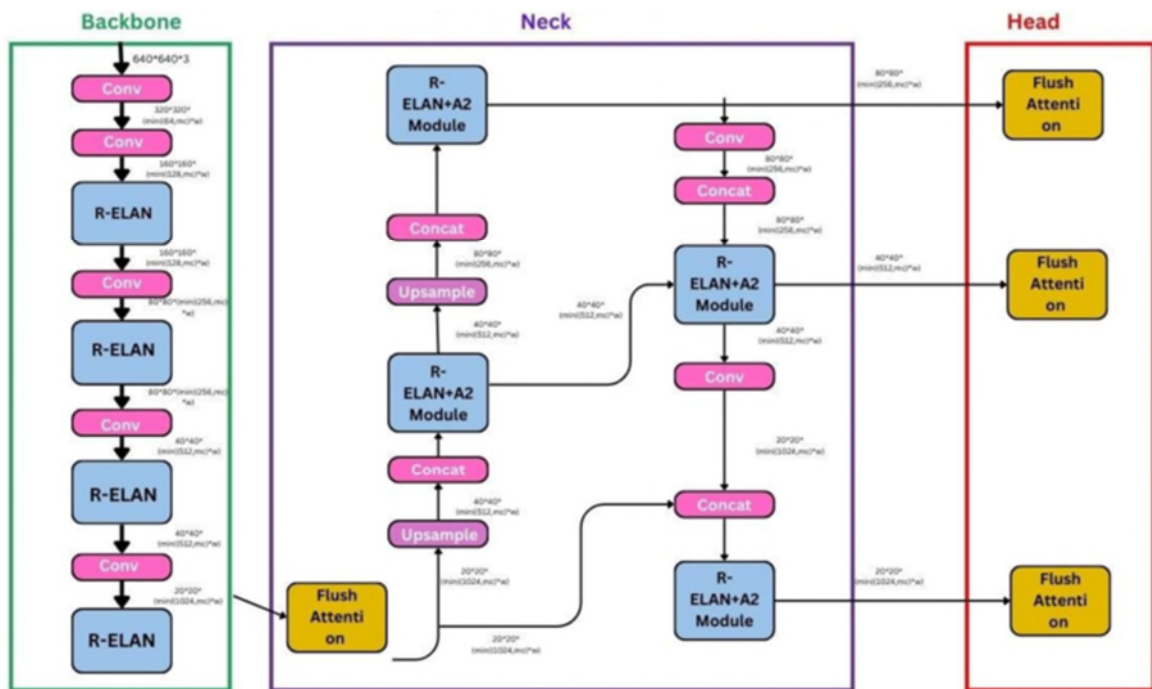


Рисунок 2.2 – Схема нейромережевої моделі [28]

Neck складається з комбінації FPN і PAN-блоків, що забезпечують вертикальне та зворотне передавання ознак. Це дозволяє моделі використовувати контекстні ознаки з різних рівнів глибини, що особливо важливо для виявлення об'єктів різного розміру, характерних для паркування, де є і великі автомобілі, і дрібні фрагменти, які частково перекриваються. Додатково у YOLOv12 інтегровано механізм Area Attention, який надає моделі змогу фокусувати обчислювальні ресурси на найбільш релевантних частинах зображення. Він ділить карту ознак на області і підсилює ті, які мають найбільше значення для виявлення об'єктів. Використання модулів FlashAttention оптимізує обчислення, зменшуючи споживання пам'яті та пришвидшуючи тренування.

Завершальним етапом є блок Head, де здійснюється фінальне прогнозування координат об'єктів, їх класифікація та визначення рівня ймовірності. У схемі видно, що на кожному масштабному рівні виконується окреме прогнозування, що дозволяє одночасно виявляти великі, середні й малі об'єкти. Модель використовує як anchor-based, так і anchor-free підходи для

гнучкості у різних сценаріях застосування. Вдосконалена функція втрат враховує не лише помилки локалізації, але й якість класифікації та впевненість у передбаченнях, що загалом підвищує точність моделі.

Окрім високої точності та швидкодії, однією з ключових переваг YOLOv12 у контексті обробки зображень паркінгу є її здатність адаптуватися до складних умов, характерних для реального середовища. Наприклад, статичні фотографії паркувальних майданчиків часто мають значну варіативність: змінне освітлення (тіні, засвітки, відблиски), присутність частково перекритих об'єктів (автомобілі, огорожі, дерева), а також різну орієнтацію та масштаби транспортних засобів. Структура YOLOv12 враховує ці чинники завдяки використанню глибоких контекстуальних ознак і багаторівневої обробки зображень, що дозволяє моделі виявляти як повністю видимі, так і частково замасковані об'єкти.

Крім того, застосування Area Attention в YOLOv12 дозволяє більш ефективно працювати з фотографіями, зробленими під різними кутами огляду. Такий підхід надзвичайно актуальний у випадках, коли камери встановлені не перпендикулярно до площини паркінгу, а під кутом, що змінює перспективу й ускладнює ідентифікацію контурів машин. Завдяки сегментації карти ознак на просторові зони і аналізу кожної з них окремо, модель може зосереджувати увагу лише на важливих зонах, ігноруючи шумові або неінформативні ділянки зображення.

Ще одним важливим аспектом є масштабованість моделі. YOLOv12 може бути адаптована до різної роздільної здатності зображень та обчислювальних можливостей системи. Це дає змогу використовувати її як на серверних обчислювальних вузлах для обробки великого обсягу фото, так і в локальних десктопних рішеннях для невеликих проєктів або пілотних систем розумного паркування. Завдяки підтримці як класичних anchor-based методів, так і anchor-free архітектур, YOLOv12 може бути швидко перенавчена на нові набори даних, отримані з конкретного паркінгу або камер відеоспостереження, що ще більше підвищує точність при локальному розгортанні.

Таким чином, кожен блок архітектури YOLOv12 на схемі виконує критично важливу функцію: від виділення ознак до їх агрегації та остаточного прогнозування. Уся архітектура оптимізована під умови обмежених ресурсів та реального часу, що робить її ідеальною для задач розпізнавання транспортних засобів на паркінгах на основі фото.

2.3 Підготовка даних для донавчання моделі

Характеристики вхідного набору даних відіграють ключову роль у формуванні точної та надійної моделі комп'ютерного зору, особливо в завданнях виявлення об'єктів на статичних зображеннях. Для реалізації завдання автоматизованого розпізнавання стану паркувальних місць (вільне або зайняте) було використано набір даних з проєкту PKLot.v1-raw, який містить реальні фотографії відкритих паркувальних майданчиків, отримані з камер відеоспостереження за різних умов зйомки. Зображення охоплюють широкий спектр варіацій освітлення (включно з сонячними ділянками, тінню та похмурими умовами), а також різноманітні кути спостереження, що моделюють реальні сценарії розміщення камер на парковках.

Особливістю цього датасету є його точна ручна розмітка: кожне паркувальне місце на зображенні позначено у форматі YOLO – зазначено координати обмежувального прямокутника (bounding box) та клас об'єкта (label), що може бути «space-occupied» або «space-empty». Завдяки єдиному формату розмітки набір даних легко інтегрується у стандартні пайплайни навчання моделей об'єктного виявлення, зокрема YOLO.

Зображення мають фіксований розмір 640×640 пікселів, що дозволяє зменшити витрати на масштабування зображень і сприяє стабільності при передобробці. Для забезпечення ефективності навчання зображення подаються на вхід моделі у вигляді порцій (batch), що дає змогу поетапно оновлювати ваги та зменшити навантаження на оперативну пам'ять. Розподіл даних відповідає загальноприйнятій пропорції: 70% складають навчальні дані (train), 20% –

валідаційні (val), і 10% – тестові (test). Датасет містить 12 439 зображень, що представляють паркувальні майданчики з різних ракурсів та в різних умовах освітлення і погоди. Кожне зображення містить кілька паркувальних місць, для яких вказано, чи є вони вільними чи зайнятими.

Фото були зроблені за допомогою камер спостереження, розміщених на справжніх паркінгах, тому датасет добре відображає реальні умови. Завдяки значній кількості зображень, а також різноманітності сцени (час доби, тіні, погодні умови), PKLot широко використовується для навчання і тестування моделей машинного навчання, зокрема нейронних мереж для автоматичного визначення зайнятості паркувальних місць. Така структура забезпечує баланс між якісним навчанням та можливістю перевірки узагальнювальної здатності моделі.

Фізична структура папок датасету чітко визначена: підкаталоги train/images та train/labels містять відповідно зображення та розмітку для тренування; valid/images і valid/labels – дані для валідації; test/images та test/labels – набір для підсумкової перевірки моделі. Файл data.yaml описує конфігурацію набору, включаючи шляхи до вибірок та список класів. Додаткові службові файли (labels.cache, README.dataset.txt, README.roboflow.txt) забезпечують швидке завантаження анотацій і супроводжувальну документацію. На рисунку 2.3 представлено структуру папок датасету.

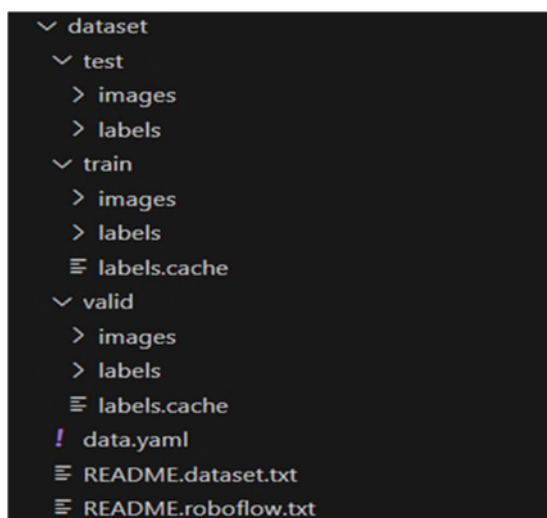


Рисунок 2.3 – Структура папок датасету [29]

На рисунок 2.4 проілюстровано приклад зображення з набору.

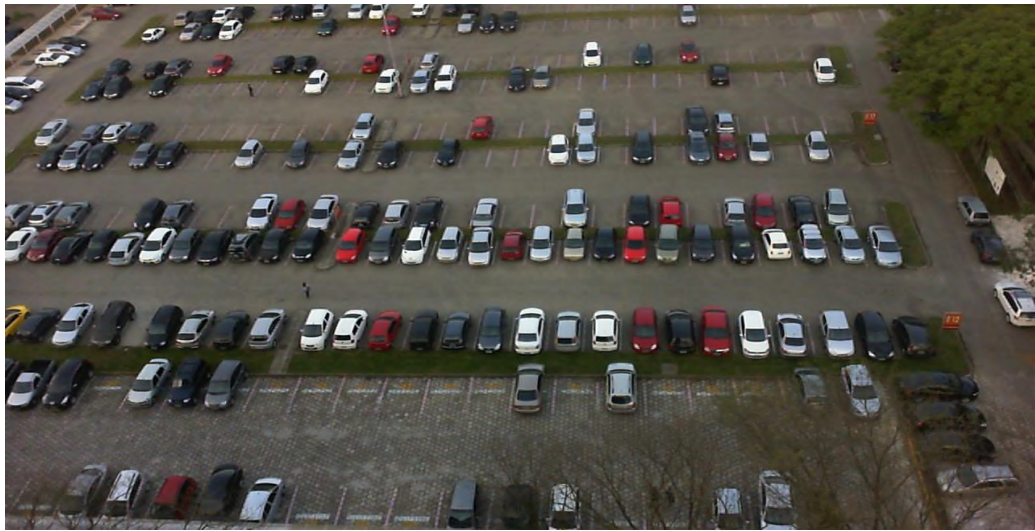


Рисунок 2.4 – Приклад зображення з набору [30]

Процес навчання моделі реалізується шляхом донавчання попередньо натренованої моделі YOLO на вказаному наборі даних. Такий підхід дає змогу використати вже наявні знання моделі про об'єкти загального призначення та переналаштувати її на специфіку паркувальних місць. Під час навчання обчислюється загальна функція втрат (loss), яка складається з кількох компонентів:

IoU Loss – мінімізує відстань між передбаченими та істинними рамками об'єктів;

Classification Loss (Cross-Entropy) – визначає правильність класифікації паркувальних місць;

Objectness Loss – оцінює ймовірність наявності об'єкта в передбаченій області;

Segmentation Loss – може бути використаний за потреби для точного виділення меж паркувальних місць.

Зазвичай, модель навчається протягом 10 – 20 епох з фіксованим learning rate. Після кожної епохи проводиться оцінка на валідаційній вибірці з розрахунком метрик точності (Accuracy), повноти (Recall), точності класифікації (Precision), середньої точності (mAP) та F1-score. Такий контроль дозволяє

виявити перенавчання, підібрати оптимальні параметри та зберегти найбільш ефективну версію моделі.

У результаті формується адаптована під конкретні вхідні умови модель YOLO, здатна з високою точністю виявляти та класифікувати паркувальні місця на статичних зображеннях. Це створює основу для впровадження точних та масштабованих систем моніторингу паркування, які здатні працювати в режимі реального часу без втрати якості в умовах зміни зовнішніх факторів.

2.4 Виявлення паркувальних місць у зображеннях

Виявлення паркувальних місць на зображеннях за допомогою нейронної мережі YOLO (You Only Look Once) базується на комплексному процесі глибокого аналізу вхідного зображення, який дозволяє одночасно виконати як локалізацію об'єктів, так і їхню класифікацію у категорії "вільне" або "зайняте". Такий підхід є надзвичайно ефективним у системах автоматизованого моніторингу паркінгів, особливо в умовах статичних зображень із камер відеоспостереження, коли необхідно забезпечити обробку великої кількості даних із високою швидкістю та точністю. На рисунку 2.5 продемонстровано схему процесу використання навченої моделі.

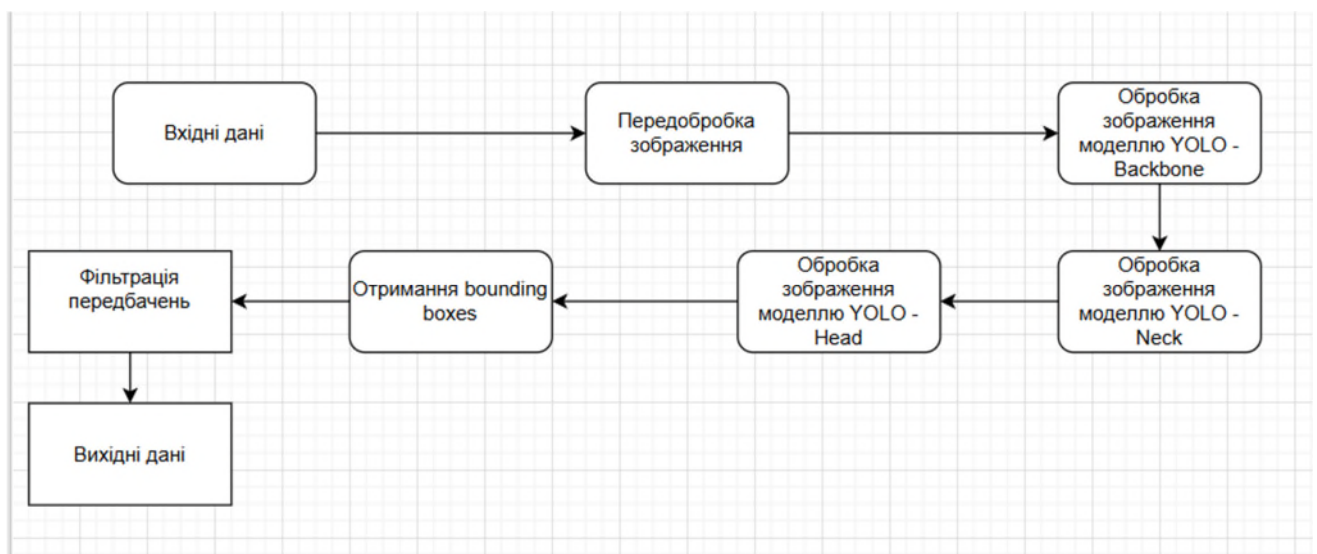


Рисунок 2.5 – Схема процесу використання навченої моделі

Крок 1. Вхідні дані (захоплення зображення камерою).

Першим елементом у процесі є камера, яка відіграє роль джерела вхідного зображення. Вона фіксується у статичному положенні таким чином, щоб охоплювати всю зону паркування з найкращим можливим оглядом. Стабільність положення камери дозволяє забезпечити незмінність перспективи та зменшити ризик виникнення артефактів через зміну ракурсу. Якість об'єктива та налаштування експозиції напряду впливають на подальшу обробку, оскільки зображення повинно бути достатньо чітким і контрастним для точного розпізнавання об'єктів.

Крок 2. Передобробка зображення.

Захоплене зображення проходить передобробку, яка є критичним етапом для підготовки вхідних даних до нейромережевої моделі. На цьому етапі зображення масштабують до фіксованого розміру, наприклад 640×640 пікселів, що відповідає очікуваному формату вхідного шару моделі YOLO. Далі виконується нормалізація значень пікселів, зазвичай у діапазон від 0 до 1, а також видаляються шуми, коригується яскравість, контраст або баланс білого. Такий підхід забезпечує уніфікацію даних і стабільну роботу моделі незалежно від умов зйомки.

Крок 3. Обробка зображення моделлю YOLO – Backbone.

Після передобробки зображення подається до нейронної мережі YOLO, починаючи з її першого внутрішнього блоку – Backbone. Цей компонент відповідає за вилучення базових просторових ознак із зображення. За допомогою кількох згорткових шарів він аналізує текстури, краї, контури, геометричні патерни, які є необхідними для ідентифікації об'єктів. Результатом є багатовимірне представлення зображення у вигляді тензорів ознак, які містять основну інформацію про його візуальний зміст.

Крок 4. Обробка зображення моделлю YOLO – Neck.

Далі ці тензори передаються в блок Neck, який виконує функцію агрегації ознак. Особливість цього етапу полягає у здатності комбінувати інформацію з кількох масштабів, що дозволяє моделі виявляти об'єкти як великих, так і малих розмірів. Neck побудований з використанням спеціальних архітектурних компонентів, таких як Feature Pyramid Network (FPN) або Path Aggregation Network (PANet), які підвищують якість передбачення на складних сценах.

Крок 5. Обробка зображення моделлю YOLO – Head.

Тут здійснюється початкове прогнозування координат обмежувальних прямокутників (bounding boxes) та визначення класу об'єктів на зображенні. Для кожного об'єкта модель також формує оцінку впевненості у передбаченні, яка надалі використовується при фільтрації результатів. Цей етап є підготовчим до формування остаточних детекцій, що відображають виявлені автомобілі на паркінгу.

Крок 6. Отримання bounding boxes.

У результаті проходження зображення через архітектуру YOLO формується масив bounding boxes. Кожен прямокутник має координати у межах зображення, відповідний клас (наприклад, “free” або “occupied”) та числове значення ймовірності. Ці дані є первинними передбаченнями, які ще не очищені від потенційних помилок.

Крок 7. Фільтрація передбачень.

Щоб забезпечити високу достовірність та точність, передбачення підлягають фільтрації. По-перше, застосовується ймовірнісний поріг: зазвичай відсікаються всі результати, впевненість у яких менша за 0.5. По-друге, впроваджується Non-Maximum Suppression (NMS), що дозволяє усунути дублікати – тобто прямокутники, які перекриваються і відповідають одному й тому ж об'єкту. Залишається лише те передбачення, яке має найвищу ймовірність.

Крок 8. Вихідні дані.

На фінальному етапі результати обробки передаються у відповідний інтерфейс користувача. У випадку десктопного або веб-застосунку координати

bounding boxes накладаються на оригінальне зображення у вигляді графічних рамок.

Загалом, модель YOLO демонструє високу ефективність у завданнях локалізації та класифікації об'єктів на зображеннях завдяки своїй здатності працювати у реальному часі без втрати точності, що робить її особливо придатною для автоматизованих систем аналізу паркувального простору, навіть за умов зміни навколишнього середовища, тіней, або складного розташування автомобілів.

2.5 Класифікація вільних паркувальних місць

У рамках прямого підходу до класифікації паркувальних місць модель глибокого навчання – зокрема, YOLO – навчається не на виявленні транспортних засобів, як це характерно для класичних схем комп'ютерного зору, а на прямій детекції та класифікації самих паркувальних місць. Основна ідея полягає в тому, що кожне паркувальне місце розглядається як окремий об'єкт на зображенні, який може належати до одного з двох класів: occupied (зайняте) або free (вільне). Таким чином, задача визначення стану місць для паркування формалізується як задача об'єктної детекції з класифікацією на два класи.

Цей підхід передбачає формування спеціального датасету, в якому вручну розмічено всі наявні паркувальні місця на кожному зображенні. Кожне місце позначається обмежувальним прямокутником (bounding box), а також присвоюється йому відповідна мітка класу (occupied або free). На рисунку 2.6 показано bounding boxes, які накладені на паркувальне місце для автоматичного визначення зайнятості паркувального місця.



Рисунок 2.6 – Bounding boxes, накладені на паркувальне місце для автоматичного визначення зайнятості паркувального місця [31]

На відміну від непрямого підходу, де спочатку визначаються автомобілі, а далі виконується аналіз просторового співвідношення між ними та заздалегідь заданими зонами паркування, у прямому методі цей проміжний етап повністю відсутній. Модель безпосередньо навчається на прикладах, у яких уже позначено, що певне паркувальне місце є вільним або зайнятим, і, відповідно, виконує класифікацію під час інференсу.

Під час роботи модель YOLO проходить звичний процес обробки зображення: спочатку за допомогою архітектури "backbone" здійснюється вилучення ознак із зображення – таких, як структура асфальту, наявність розмітки, форма тіней, колірні відмінності або контури авто. Потім ці ознаки передаються на рівень "neck" (зазвичай це FPN або PAN), який відповідає за агрегацію ознак на різних масштабах. На завершальному етапі "head" модель формує кінцеві передбачення: координати кожного паркувального місця на зображенні, клас об'єкта (вільне чи зайняте) та рівень впевненості (confidence score).

Результатом роботи системи є набір прямокутників з відповідними класами. На виході ми отримуємо повну інформацію щодо кожного паркувального місця і його місцезнаходження на кадрі, поточний стан і ймовірність правильної класифікації. Таким чином, інтерфейс системи може одразу відобразити кольорову схему паркінгу або числову аналітику, не виконуючи додаткових обчислень чи аналізу положення авто.

Однією з ключових переваг такого підходу є простота використання та швидкість обробки. Оскільки модель видає вже готові передбачення про стан паркувального місця, відпадає необхідність у постобробці, геометричному аналізі або перетині bounding box транспортних засобів з координатами місць. Це особливо корисно в умовах обмежених обчислювальних ресурсів або при потребі отримання результату в режимі реального часу. Також прямий підхід добре підходить для сценаріїв, де камера зафіксована під незмінним кутом, наприклад, на шлагбаумі, фасаді будівлі або в дроні, що регулярно пролітає заданим маршрутом.

Однак існують і певні виклики. Передусім, якість класифікації сильно залежить від якості та обсягу навчальних даних. Паркувальні місця повинні бути точно розмічені, а стан – коректно позначений на всіх прикладах. Крім того, модель, натренована на одному конкретному паркінгу чи ракурсі, може демонструвати знижену ефективність при перенесенні на інші локації або змінених умовах освітлення, перспективи тощо. Це вимагає або регулярного донавчання моделі, або використання гнучкіших архітектур, що здатні до адаптації.

Варто також зазначити, що прямий підхід у деяких випадках може виявитися більш точним, ніж непрямий. Наприклад, якщо автомобіль лише частково заходить на місце або має нестандартні розміри, то аналіз просторового перекриття (в непрямому методі) може дати хибний результат. Натомість модель, натренована безпосередньо на прикладах таких ситуацій, зможе навчитися їх правильно інтерпретувати – тобто вважати місце зайнятим або вільним, навіть якщо немає повного перекриття.

Загалом, прямий підхід класифікації паркувальних місць є ефективним методом, який поєднує точність, швидкодію та простоту в реалізації.

2.6 Метрики оцінювання ефективності моделі

Розробка систем автоматичного виявлення та класифікації паркувальних місць є однією з ключових задач у сфері інтелектуальних транспортних систем і «розумних» міст. Оцінювання якості таких систем – це не просто формальність, а важливий етап, що дозволяє не лише перевірити правильність роботи алгоритмів, а й виявити їхні обмеження та області для вдосконалення.

У процесі розробки та тестування моделей класифікації широко використовують стандартні метрики, кожна з яких дає власний погляд на ефективність системи. Вибір правильної метрики залежить від конкретних вимог і контексту застосування системи.

Ассурасу (загальна точність) – найпростіший показник, що характеризує загальну кількість коректних передбачень відносно всіх передбачень. Проте, у випадках із дисбалансом класів, наприклад коли більшість паркувальних місць зайняті, ассурасу може вводити в оману. Модель може видавати дуже високі значення цього показника, навіть якщо фактично вона погано визначає рідкісні вільні місця [32].

Precision (точність) відображає, наскільки коректними є позитивні передбачення моделі, тобто яка частина передбачених вільних місць дійсно вільна. В системах паркування це критично, оскільки помилкове повідомлення про вільність зайнятого місця може призвести до неприємних ситуацій для користувача. Висока precision свідчить про те, що користувачеві можна довіряти у питаннях наявності вільних місць [33].

Recall (повнота) визначає здатність системи не пропустити жодного вільного місця. В умовах обмеженої кількості паркувальних місць це надзвичайно важливо, адже пропущене вільне місце – це втрата потенційної можливості для паркування. Висока recall вказує на те, що модель максимально "обережна" у виявленні вільних місць, навіть якщо це іноді призводить до помилкових сигналів [34].

F1-Score є збалансованим показником, який інтегрує точність і повноту, надаючи цілісну картину якості моделі. В ситуаціях, коли потрібно досягти компромісу між двома протилежними завданнями – не пропустити вільне місце і не помилково повідомляти про зайнятість, F1-Score служить хорошим орієнтиром [35].

У контексті паркувальних систем важливо враховувати специфіку даних і наслідки помилкових класифікацій. Наприклад, помилка типу false positive (хибне повідомлення про вільність місця, що зайняте) може призвести до збитків часу для водія, зниження довіри до системи і навіть до аварійних ситуацій на паркувальних майданчиках. Помилка false negative (непомічене вільне місце) теж є небажаною, оскільки призводить до зниження загальної пропускної здатності паркінгу.

Відповідно, під час розробки системи необхідно визначити пріоритетність тих чи інших метрик залежно від конкретного сценарію застосування. У великих міських паркінгах, де кількість вільних місць часто обмежена, важливішим буде максимізація recall. У приватних або корпоративних паркінгах, де користувачі очікують максимальної точності, критичнішою може бути precision. Для глибшого розуміння роботи моделі та діагностики проблем застосовують різноманітні графічні інструменти. Вони дозволяють не лише побачити цифри, а й зрозуміти, де і чому модель може помилятися.

Матриця невідповідностей (Confusion Matrix) є основним інструментом для візуалізації результатів класифікації. Вона відображає, скільки разів модель класифікувала паркувальні місця правильно і де саме трапляються помилки. Матриця допомагає швидко ідентифікувати, чи модель віддає перевагу певному класу, чи має тенденцію до певного типу хибних передбачень. Наприклад, якщо в матриці видно багато false positives, це сигнал про те, що модель часто помилково вважає зайняті місця вільними. Приклад Confusion Matrix на рисунку 2.7.

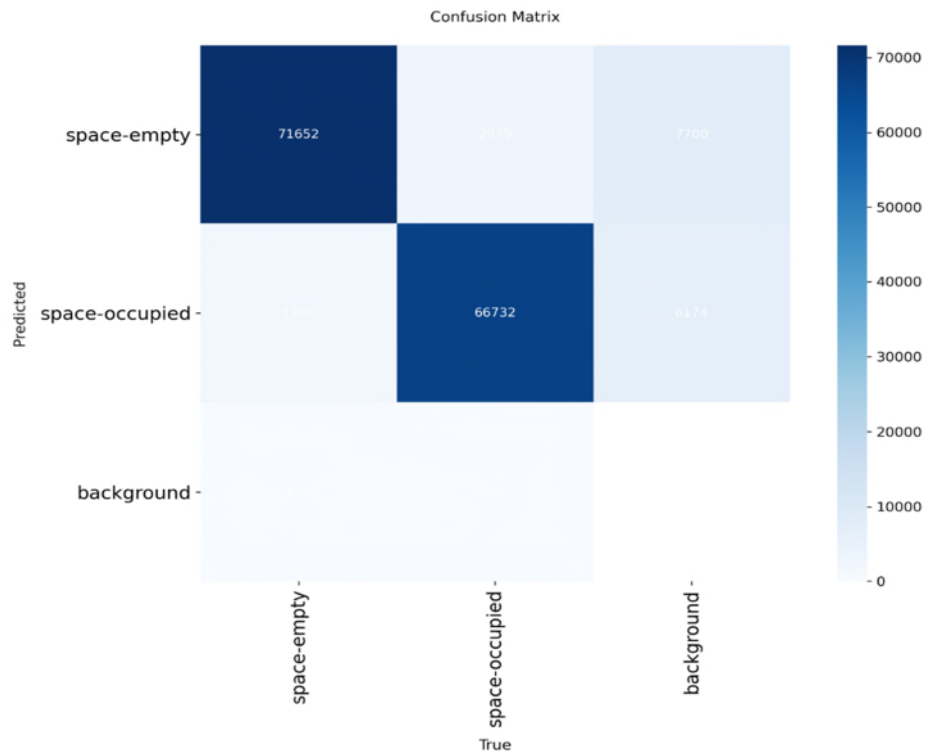


Рисунок 2.7 – Приклад Confusion Matrix [36]

Дана матриця демонструє результати класифікації трьох класів: вільні паркувальні місця (*space-empty*), зайняті паркувальні місця (*space-occupied*) та фон (*background*). Модель доволі впевнено справляється із задачею, з високою точністю класифікуючи як вільні, так і зайняті місця. Зокрема, 71 652 вільних місця були класифіковані правильно, при цьому 2 979 з них були помилково визначені як зайняті, а 7 700 – як фон. Для зайнятих місць 66 732 були правильно класифіковані, 1 986 – хибно як вільні, а 6 174 – як фон. Клас "background" також здебільшого не викликав труднощів, що видно з незначної кількості помилок: 174 випадки помилкової класифікації як вільні місця та 220 – як зайняті.

Precision-Recall крива дозволяє аналізувати зміну точності і повноти при варіюванні порогу класифікації. Завдяки цьому графіку можна підібрати оптимальний поріг, який забезпечить бажаний баланс між точністю і повнотою відповідно до потреб системи.

ROC-крива (Receiver Operating Characteristic Curve) використовується для оцінки здатності моделі розрізняти позитивні і негативні класи. Площа під ROC-кривою (AUC) є узагальненим показником якості моделі незалежно від вибору

порогу. Значення AUC, близьке до 1, свідчить про високий рівень розрізнення класів. Приклад ROC – кривої на рисунку 2.8.

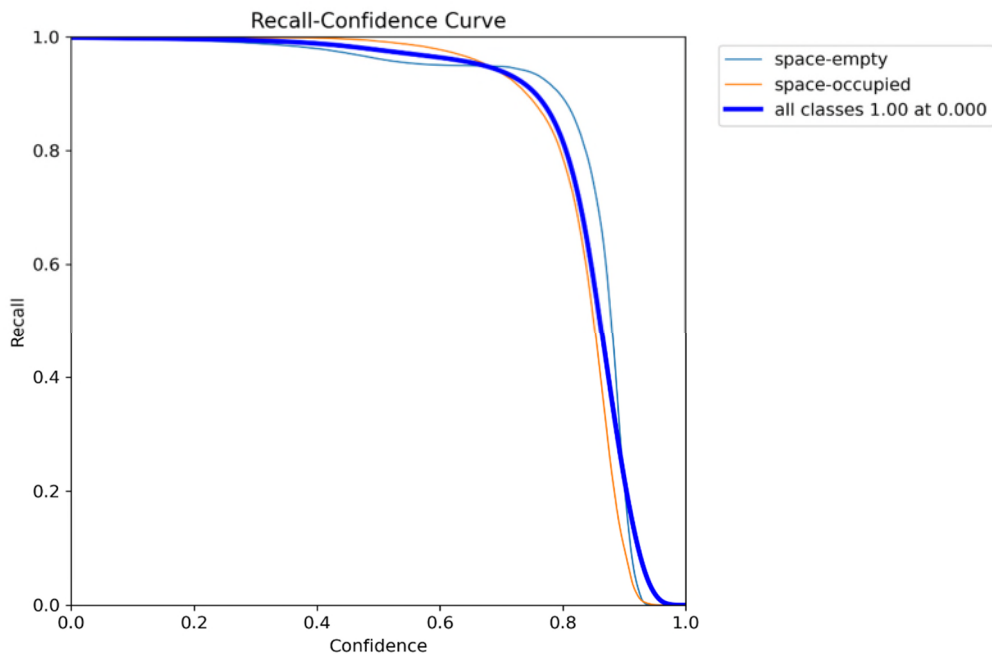


Рисунок 2.8 – Приклад ROC – кривої [37]

Криві показують, як змінюється повнота (recall) в залежності від порогу впевненості, починаючи з нульового значення до максимального (1.0).

На низьких значеннях confidence (0.0–0.6) повнота залишається максимальною (приблизно 1.0), що свідчить про те, що модель виявляє майже всі об'єкти. Однак із зростанням порогу впевненості повнота різко знижується, особливо після позначки 0.85, коли модель починає ігнорувати менш впевнені передбачення. Це типовий ефект, коли зростання впевненості супроводжується зменшенням чутливості.

Клас «space-empty» має трохи вищу повноту за всіх рівнів упевненості порівняно з «space-occupied», що свідчить про дещо кращу здатність моделі виявляти вільні місця. У середньому ж модель демонструє високу стабільність для обох класів на великому проміжку значень довіри. Це означає, що її можна ефективно використовувати для детекції стану паркувальних місць із контролем за точністю шляхом коригування порогу впевненості.

При оцінці системи важливо застосовувати не лише кількісні метрики, а й розглядати поведінку моделі на реальних даних. Наприклад, корисно проводити огляд помилок на окремих зображеннях або відео, аналізувати випадки, коли система плутає об'єкти чи не помічає паркувальні місця через затінення, погане освітлення або часткове перекриття автомобілем.

Якісне оцінювання роботи системи класифікації паркувальних місць потребує комплексного підходу, який включає як кількісні метрики, так і візуальний аналіз. Поєднання метрик accuracy, precision, recall і F1-score з інформативними візуалізаціями, такими як матриця невідповідностей, Precision-Recall і ROC-криві, дозволяє комплексно оцінити модель і вибрати оптимальні налаштування для конкретних задач.

2.7 Висновки до розділу 2

Розглянуто концепцію визначення вільних паркувальних місць на відкритих стоянках із використанням нейромережових технологій. Наведено структурну схему методу, що описує етапи обробки зображення: від виявлення паркомісць до класифікації їхнього стану. Описано архітектуру моделі, здатної ефективно розпізнавати об'єкти в умовах різного освітлення та положення камер.

Проаналізовано підходи до автоматичного визначення стану паркувальних місць і метрики оцінювання якості моделі – точність, повнота та продуктивність.

Сформульовано завдання реалізувати програмний застосунок, що автоматизує процес обробки зображень і визначення вільних місць. Для оцінки ефективності передбачено проведення функціонального дослідження з використанням реальних або тестових даних та аналіз отриманих метрик.

У результаті окреслено основні компоненти для побудови ефективної системи виявлення вільних паркувальних місць на основі аналізу зображень.

Розділ 3 Особливості реалізації та результати тестування методу визначення вільних місць на відкритих паркуваннях

3.1 Особливості розробки програмного застосунку для імплементації запропонованого методу

3.1.1 Засоби розробки програмного застосунку

У сучасних умовах стрімкого розвитку технологій розумних міст особливої актуальності набувають системи інтелектуального моніторингу паркувального простору. Зважаючи на обмеженість інфраструктури та постійне зростання кількості транспортних засобів, ефективне управління паркуванням стає важливою інженерною задачею. У зв'язку з цим доцільним є використання автоматизованих систем, здатних у режимі реального часу визначати наявність вільних або зайнятих місць на відкритих паркувальних майданчиках. Для побудови такої системи було обрано підхід на основі методів комп'ютерного зору та глибокого навчання.

Мова програмування Python є головним інструментом, який використовується в розробці програмного забезпечення для реалізації методу виявлення вільних паркувальних місць. Python – це високорівнева мова з відкритим кодом, яка відзначається своєю простотою, лаконічним синтаксисом і широкою підтримкою середовища розробників. Її зручність полягає в тому, що вона поєднує простоту написання коду з потужними можливостями, що дозволяє ефективно створювати як прототипи, так і великі системи. Вона підтримує багато парадигм програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне програмування, що дає можливість будувати гнучкі та масштабовані архітектури [38]. Завдяки тому, що Python є інтерпретованою мовою, вона дозволяє швидко запускати та тестувати частини коду, що прискорює розробку й полегшує налагодження. Крім того, вона є кросплатформною, тобто написані на ній програми можуть запускатися на Windows, Linux або macOS без потреби у зміні коду, що особливо важливо для систем, які можуть розгортатися в різноманітних умовах. У випадку дослідження паркувальних місць Python

використовується для створення всього програмного середовища, керування обробкою зображень і відео, організації потоку даних, а також для інтеграції моделі нейронної мережі, що виконує детекцію.

Фреймворк PyTorch є фундаментом для створення нейронних мереж, які застосовуються для аналізу відеопотоку і виявлення об'єктів. PyTorch було розроблено дослідницьким підрозділом компанії Facebook і він швидко став одним з провідних фреймворків у сфері глибокого навчання завдяки гнучкості, продуктивності та активній спільноті. Основною особливістю PyTorch є використання динамічних обчислювальних графів – це означає, що граф моделі формується під час виконання програми, що робить фреймворк зручним для експериментів, змін архітектури моделі та побудови нових методів. PyTorch також має повноцінну підтримку роботи з графічними процесорами, що дозволяє проводити навчання на великих обсягах відеоданих значно швидше, ніж на звичайних процесорах [39]. У цьому проєкті PyTorch використовується як базова технологія для створення моделі обробки зображень, побудованої на архітектурі YOLO, зокрема через платформу Ultralytics. Його модульна структура дає змогу зручно підключати потрібні компоненти, як-от функції втрат, оптимізатори, активаційні функції, а також організовувати обробку вхідних та вихідних даних, що є ключовим для високоточних задач комп'ютерного зору.

Ultralytics – це розширення над фреймворком PyTorch, яке спеціалізується на роботі з моделями YOLO (You Only Look Once) – одними з найефективніших архітектур для задач виявлення об'єктів у зображеннях і відео. Ця платформа розроблена з урахуванням зручності користування і дозволяє запускати повний цикл роботи з моделлю: від налаштування, навчання та валідації до використання у реальному часі. Ultralytics особливо корисна в дослідженні, оскільки дає змогу навчити модель на власному наборі даних, який відповідає специфіці завдання – виявлення вільних місць на відкритих паркінгах. Платформа має гнучкий механізм конфігурації параметрів через YAML-файли, що дозволяє швидко змінювати архітектуру моделі, оптимізатори, параметри навчання, стратегії підбору гіперпараметрів тощо [40]. Також Ultralytics має вбудовані засоби для

візуалізації результатів, ведення статистики тренування, використання метрик продуктивності, таких як mAP, точність, recall, що дає змогу оцінювати ефективність моделі та обґрунтовувати результати. Важливо й те, що Ultralytics підтримує інтеграцію з WandB (Weights and Biases) – спеціалізованою платформою для моніторингу навчання моделей у хмарному середовищі. Таким чином, Ultralytics є повноцінним інструментом, який дозволяє створити, налагодити і впровадити ефективну модель детекції об'єктів, що ідеально підходить для завдань в рамках запропонованого дослідження [41].

OpenCV, або Open Source Computer Vision Library, є спеціалізованою бібліотекою для комп'ютерного зору, яка включає понад дві з половиною тисячі оптимізованих функцій для аналізу зображень і відео. Вона активно використовується в академічних і промислових проєктах завдяки своїй стабільності, високій швидкодії та широким функціональним можливостям. У межах цього дослідження OpenCV відіграє ключову роль у попередній обробці відео: вона дозволяє зчитувати відеопотік з камер, витягувати окремі кадри, конвертувати їх у потрібний формат, змінювати розмір, фільтрувати шуми, виділяти контури, зони інтересу та виконувати інші перетворення. Це необхідно для підготовки даних до подачі у модель, яка працює на основі PyTorch та Ultralytics [42]. Крім цього, OpenCV дозволяє візуалізувати результати детекції – наприклад, накладати прямокутники на знайдені паркувальні місця, відображати класифікаційні мітки, рахувати кількість вільних місць тощо. Її можливості зчитування різноманітних відеоформатів, сумісність з Python і висока продуктивність дають змогу обробляти відео в реальному часі, що є важливою вимогою до систем автоматичного моніторингу паркінгів.

Customtkinter – це вдосконалений інтерфейсний фреймворк, що базується на стандартній бібліотеці tkinter, але пропонує сучасний дизайн і ширші можливості стилізації графічного інтерфейсу. Основною перевагою є підтримка темного режиму, округлених кнопок, слайдерів, випадаючих списків, прогрес-барів та інших елементів з покращеною візуальною привабливістю, що не передбачено у класичному tkinter [43]. Це дозволяє створювати сучасні та

інтуїтивно зрозумілі GUI-додатки без необхідності використання складних сторонніх інструментів чи веб-фреймворків.

У межах даного проєкту `customtkinter` застосовується для побудови інтерактивного графічного інтерфейсу, за допомогою якого користувач може завантажити зображення, встановити параметри моделі (розмір, поріг впевненості, IoU), запустити детекцію, переглянути результат у візуальному вигляді, а також ознайомитися з аналітичними метриками точності. Бібліотека надає зручні засоби розміщення віджетів, обробки подій (наприклад, натискання кнопок) та динамічного оновлення вмісту інтерфейсу. Завдяки цьому вдалося реалізувати адаптивний інтерфейс, який не лише виконує роль «обгортки» навколо моделі, але й забезпечує зручну взаємодію з користувачем та підвищує загальну ефективність системи.

Таким чином, поєднання перелічених засобів дозволяє реалізувати повноцінну метод виявлення вільних паркувальних місць. Обрана архітектура забезпечує модульність і масштабованість, що дозволяє при потребі змінювати складові проєкту, адаптувати модель до інших умов або додавати нову функціональність.

3.1.2 Програмна реалізація та інтерфейс користувача

Розроблений метод для виявлення зайнятих та вільних паркувальних місць базується на використанні глибокої нейронної мережі YOLOv12, бібліотеки OpenCV та фреймворку PyTorch, і реалізований за допомогою мови програмування Python у середовищі Visual Studio Code. Цей застосунок використовує попередньо зібрані або офіційні датасети, а натомість працює з окремими зображеннями паркувальних майданчиків, що дозволяє продемонструвати ключові можливості аналізу статичних сцен. Уся логіка зосереджена на обробці окремих знімків з відеокамер або збережених фото, без потокової обробки відео.

У структурі проєкту виявлення зайнятих і вільних паркувальних місць, створеного на основі моделі YOLO , яка показана на рисунку 3.1 , ключову роль відіграють такі файли, як `app.py`, `train.py`, а також натреновані ваги моделі. Кожен із цих компонентів виконує окрему важливу функцію, забезпечуючи повний цикл роботи системи: від навчання моделі до інтерактивної обробки зображень через графічний інтерфейс.

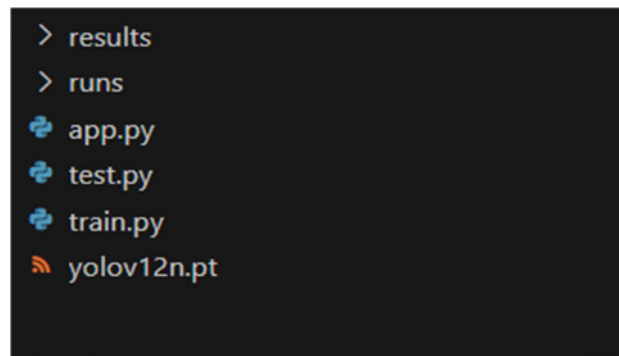


Рисунок 3.1 – Структура проєкту

Окрему роль відіграють натреновані моделі у вигляді файлів `.pt` (наприклад, `best.pt`). Ці файли містять збережені ваги нейронної мережі після завершення навчання. Вони підвантажуються під час запуску застосунку і використовуються для обробки зображень у реальному часі.

Структура проєкту також включає стандартну директорію `runs/`, яка створюється бібліотекою `ultralytics` під час тренування чи інференсу. У ній зберігаються результати обробки, логування параметрів запуску, графіки метрик, візуалізації та інша діагностична інформація.

На рівні структури програмного забезпечення застосунк поділяється на кілька взаємопов'язаних компонентів: модуль завантаження та попередньої обробки зображення, модуль виявлення об'єктів, модуль фільтрації результатів, та візуалізаційний модуль для накладення інформації про паркувальні місця на вихідне зображення. Варто детально розглянути кожен із цих компонентів.

На першому етапі користувач запускає програму це показано на рисунку 3.2, яка ініціалізує попередньо натреновану модель YOLOv12, збережену у вигляді файлу `.pt` (PyTorch weights). Для роботи з цією моделлю використовується

глибокий навчальний фреймворк PyTorch, який забезпечує ефективне завантаження ваг, обчислення тензорів на CPU або GPU та підтримує функціонал інференсу. Завдяки PyTorch, модель може працювати локально, без потреби в онлайн-сервісах, забезпечуючи повну автономність системи. У верхній частині інтерфейсу розташовані основні параметри налаштування обробки зображень. Параметр «Розмір зображення (imgsz)» відповідає за масштабування вхідного зображення до вказаного розміру перед його передачею до моделі. У даному випадку значення встановлено на 960 пікселів, що дозволяє досягти компромісу між швидкістю обробки та точністю розпізнавання.

Параметр «Поріг довіри (conf)» визначає мінімальну ймовірність, з якою модель повинна визнати об'єкт як дійсно виявлений. При значенні 0.1 система фільтрує об'єкти з меншим рівнем довіри, що дозволяє уникнути хибнопозитивних результатів.

Параметр «IoU (Intersection over Union)» використовується для визначення ступеня перекриття між прямокутниками, що позначають знайдені об'єкти. Якщо перекриття перевищує заданий поріг, система вважає, що йдеться про один і той самий об'єкт. Це дозволяє уникнути дублювання результатів під час виявлення.

Також у верхній частині розміщені кнопки «Обрати інше зображення» для завантаження зображення з локального пристрою та «Відкрити галерею» для можливості вибору фото з галереї. При натисканні на цю кнопку користувач бачить у вікні застосунку акуратно розташовані мініатюри всіх доступних фотографій із цієї папки. Галерея реалізована у вигляді сітки, де зображення представлені у вигляді кнопок із маленькими прев'ю, що дозволяє швидко орієнтуватися у вмісті. Кожне мініатюрне зображення є інтерактивним – при кліці на нього відбувається миттєвий перехід у режим детекції для вибраного фото.



Рисунок 3.2 – Головний інтерфейс застосунку

На рисунку 3.3 завантажується зображення паркування, яке користувач обирає самостійно. Зображення завантажується у вигляді об'єкта бібліотеки OpenCV, після чого здійснюється його попередня обробка. Цей етап включає зміну розміру зображення до відповідних параметрів (наприклад, 640×640 пікселів), нормалізацію пікселів та перетворення до потрібного формату тензора, який подається на вхід нейронної мережі. Саме ця обробка забезпечує уніфікацію вхідних даних і гарантує правильну роботу моделі незалежно від початкових характеристик зображення.

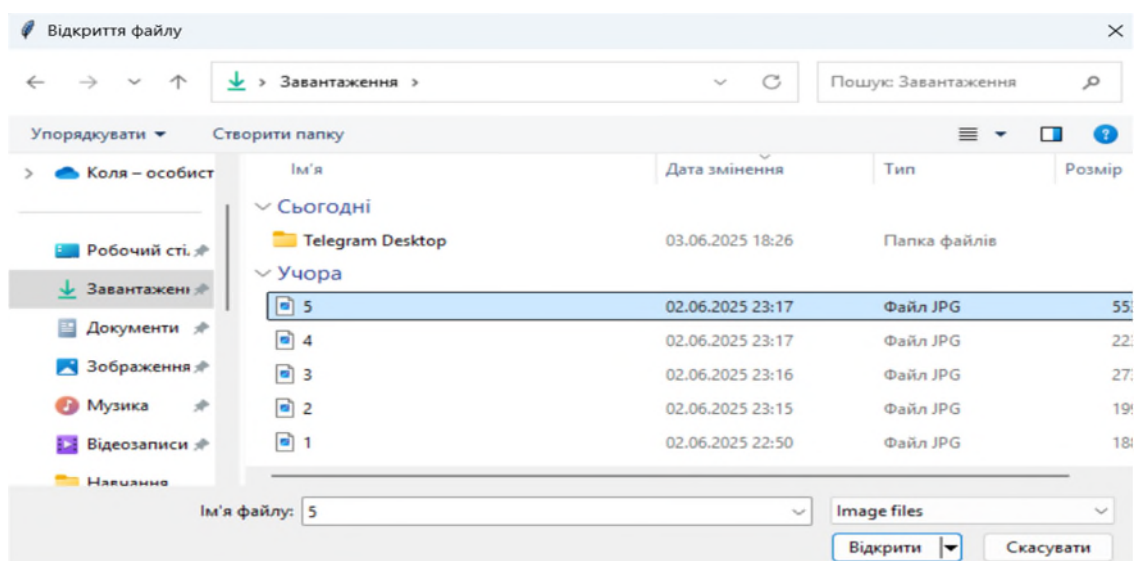


Рисунок 3.3 – Завантажування зображення паркування

Після подачі зображення у модель YOLOv12, запускається процес інференсу – передбачення. YOLO миттєво сканує зображення, розбиваючи його на сітку і для кожної комірки формує передбачення щодо наявності об'єктів. У нашому випадку – це транспортні засоби, які автоматично визначаються як об'єкти, що потенційно займають місця на паркінгу. Результатом є набір обмежувальних прямокутників (bounding boxes), які обрамлюють ці транспортні засоби та супроводжуються числовим значенням ймовірності виявлення.

Ці передбачення можуть містити помилкові або зайві виявлення. Тому в подальшому виконується етап постобробки. Він передбачає відсікання результатів із ймовірністю меншою за встановлений поріг (наприклад, 0.5), а також застосування методу Non-Maximum Suppression, який дозволяє уникнути накладання декількох рамок на один і той самий об'єкт. У результаті залишається лише найбільш достовірна інформація щодо об'єктів на зображенні.

Наступним кроком є візуалізація. У цьому модулі застосовується функціонал OpenCV для побудови прямокутників навколо кожного виявленого автомобіля, а також накладання текстових міток, які визначають статус кожного паркувального місця. Оскільки програма не містить етапу сегментації чи попереднього маркування місць, класифікація у "вільне" чи "зайняте" здійснюється за наявністю об'єкта в межах паркувального місця. Це дозволяє навіть за відсутності еталонної карти місць демонструвати функціональність розпізнавання зайнятих позицій.

Окремо слід наголосити, що застосунок реалізований у максимально автономному вигляді – він не залежить від інтернет-з'єднання, хмарних API чи зовнішніх баз даних. Усі обчислення відбуваються локально, що робить його придатним для вбудованих рішень або демонстраційних стендів. Головна цінність такої реалізації полягає в її простоті, доступності вихідного коду для подальшої адаптації, а також у можливості масштабування до потокового відео при зміні лише кількох рядків коду.

На рисунку 3.4 наведено приклад результату роботи програми, а саме вихідне зображення паркінгу зі вставленими рамками навколо автомобілів та текстовими позначеннями. Видно, як кожен виявлений об'єкт має чітко окреслений контур, що свідчить про правильне функціонування YOLOv12 навіть у реальних, неідеальних умовах освітлення й ракурсу.

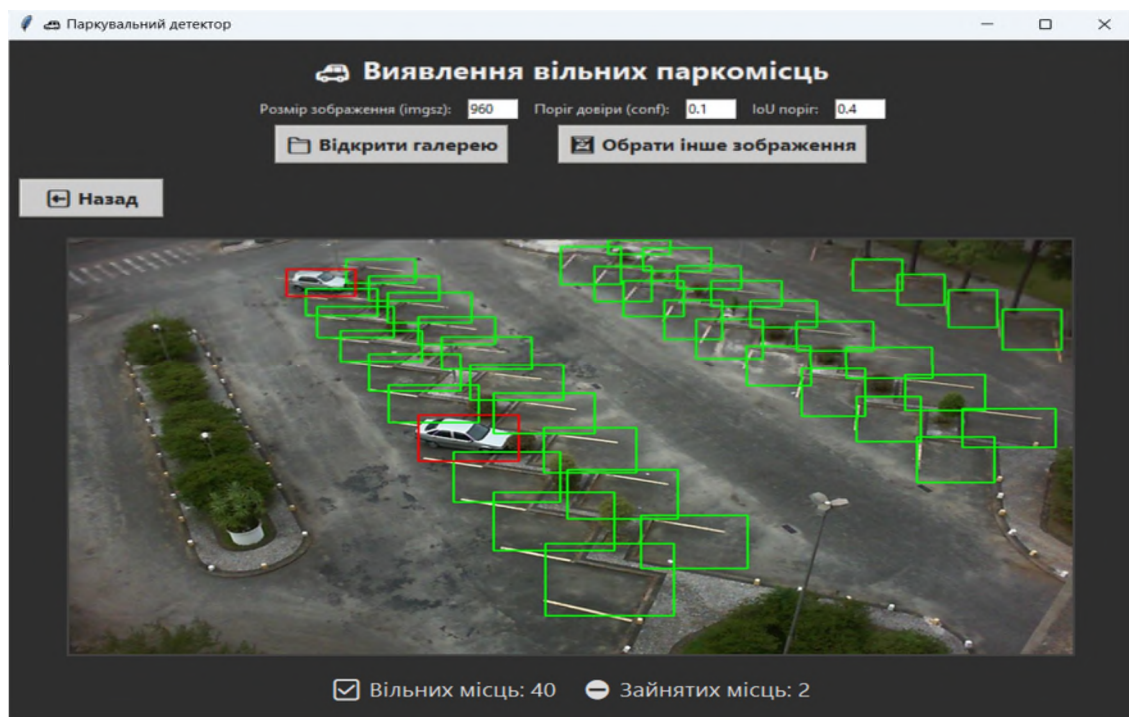


Рисунок 3.4 – Результат роботи програми

Завдяки використанню сучасної моделі YOLOv12 та бібліотеки OpenCV, досягнуто високої точності, швидкості та візуальної наочності виводу, що дозволяє інтегрувати таке рішення у більші автоматизовані системи моніторингу або паркувальні сервіси.

3.2 Результати тестування методу визначення вільних паркувальних місць

3.2.1 Результати донавчання моделі YOLO на навчальній вибірці

У процесі створення методу автоматизованого розпізнавання зайнятості паркувальних місць особливу увагу було приділено етапу донавчання моделі, з

подальшою детальною оцінкою її ефективності за допомогою ключових метричних показників. Завдання полягало у класифікації кожного паркувального місця за двома класами: «вільне» та «зайняте», що є класичною задачею бінарної класифікації, але в умовах об'єктної детекції. Для цього було використано сучасний підхід до тренування моделей на основі штучних нейронних мереж з глибоким навчанням.

Після попереднього тренування, модель була донавчена на 70% від повного обсягу доступного датасету. З метою забезпечення контролю якості навчання та запобігання перенавчанню (overfitting). Оцінка якості роботи моделі здійснювалась на основі чотирьох базових метрик, що найчастіше використовуються в машинному навчанні: точність класифікації (Accuracy), точність позитивного класу (Precision), повнота (Recall) та F1-міра. Кожна з цих метрик була не лише чисельно обчислена для кожної епохи донавчання, але й візуалізована за допомогою графіків, побудованих з використанням бібліотек Matplotlib та Seaborn.

У таблиці 3.1 показано метрики при донавчанні моделі.

Таблиця 3.1 – Результати метрик

epoch	time	train/box_loss	train/cls_loss	train/df_loss	metrics/precision (B)	metrics/s/recall (B)	metrics/mAP50(B)	metrics/mAP50-95(B)
1	595.454	1.44364	1.16132	1.02965	0.94275	0.96379	0.97014	0.71836
2	1169.56	0.99452	0.60775	0.89387	0.95723	0.9608	0.9862	0.79741
3	1758.42	0.86011	0.52863	0.86343	0.96094	0.96786	0.98772	0.81821
4	2365.52	0.77814	0.48704	0.84752	0.96574	0.96774	0.98995	0.8216
5	2984.92	0.71034	0.45688	0.83422	0.96511	0.97333	0.99135	0.8383

Оскільки модель виконує не просту бінарну класифікацію, а детекцію об'єктів, традиційне поняття Accuracy не використовується безпосередньо. Натомість, для оцінки якості детекції використовується метрика mean Average Precision при пороговому коефіцієнті $\text{IoU} \geq 0.5$). Ця метрика дає уявлення про середню точність виявлення об'єктів по всіх класах (в нашому випадку класів два:

вільне та зайняте місце). На графіку mean Average Precision зміни по епохах помітно чітко стабільне зростання з приблизно 97% на першій епосі до понад 99% на п'ятій, що демонструє високу здатність моделі правильно детектувати паркувальне місце та класифікувати їх без систематичних помилок. Цей ріст відбувався без значних стрибків чи падінь, що свідчить про добре підібрані гіперпараметри навчання та якісну аугментацію даних.

Precision є надзвичайно важливою метрикою у випадках, коли критично зменшити кількість помилкових позитивних спрацювань – тобто таких, коли система помилково визначає вільне місце як зайняте. Ця помилка потенційно може змусити водія проігнорувати доступне місце, вважаючи його зайнятим. Протягом усього донавчання, значення метрики Precision залишалося на високому рівні, починаючи з 94% на першій епосі і поступово зростаючи до 96.5% на п'ятій. Такий результат демонструє, що модель здатна точно визначати, коли місце дійсно зайняте, і практично не допускає хибних визначень. Графік метрики Precision можна побачити на рисунку 3.5.

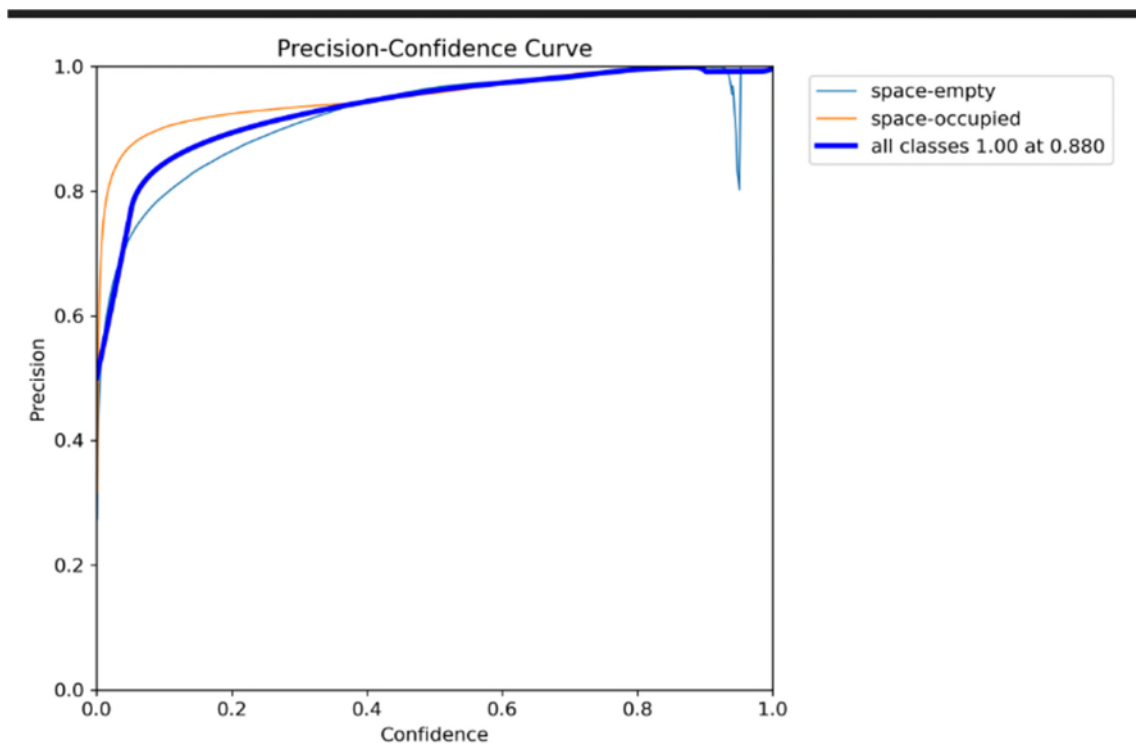


Рисунок 3.5 – Графік метрики Precision

Recall визначає здатність моделі виявляти всі наявні позитивні об'єкти – у нашому випадку, всі реально зайняті місця. Його значення особливо важливе в контексті паркінгу, де критично важливо, щоби система не пропускала зайняті місця, позначаючи їх як вільні, адже це може призвести до неправильного інформування водія. Значення Recall зросло з 96.3% на першій епісі до 97.3% на п'ятій, що вказує на виняткову здатність моделі знаходити практично всі зайняті місця без суттєвих пропусків. Також слід відзначити, що підвищення Recall відбувалося пропорційно до Precision, тобто модель не йшла на компроміси між цими показниками. На рисунку 3.6 продемонстровано графік метрики Recall.

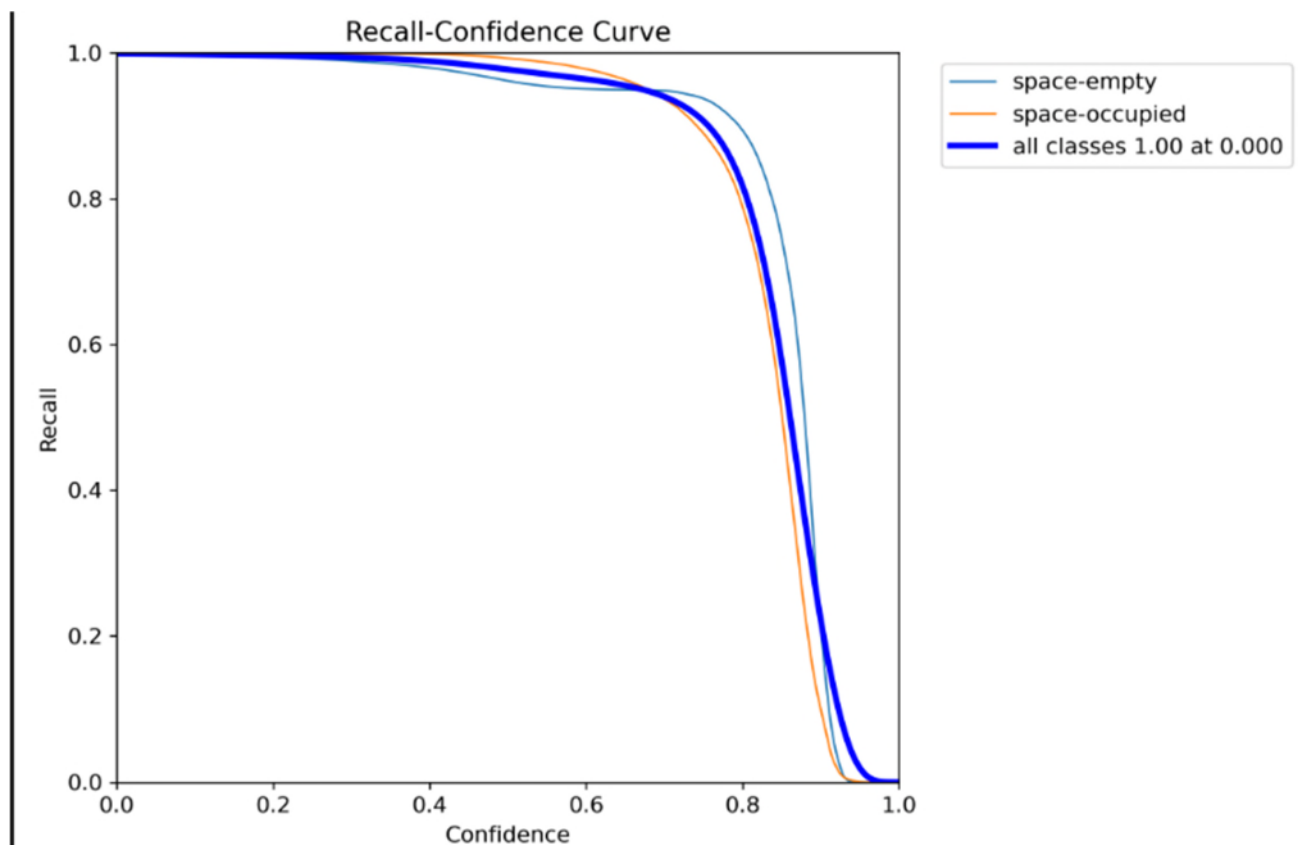


Рисунок 3.6 – Графік метрики Recall

F1-score поєднує в собі Precision та Recall і дозволяє оцінити баланс між точністю та повнотою. Важливість цієї метрики зростає тоді, коли обидва показники мають високе значення і важко визначити, який із них важливіший. За результатами аналізу видно, що F1-міра зростала протягом усього навчання і на п'ятій епісі досягала приблизно 96%, що є відмінним результатом для бінарної

класифікації в умовах детекції. Цей показник є ключовим доказом збалансованої роботи моделі без перекосів у бік надмірної чутливості чи надмірної обережності. На рисунку 3.7 продемонстровано графік метрики F1-score.

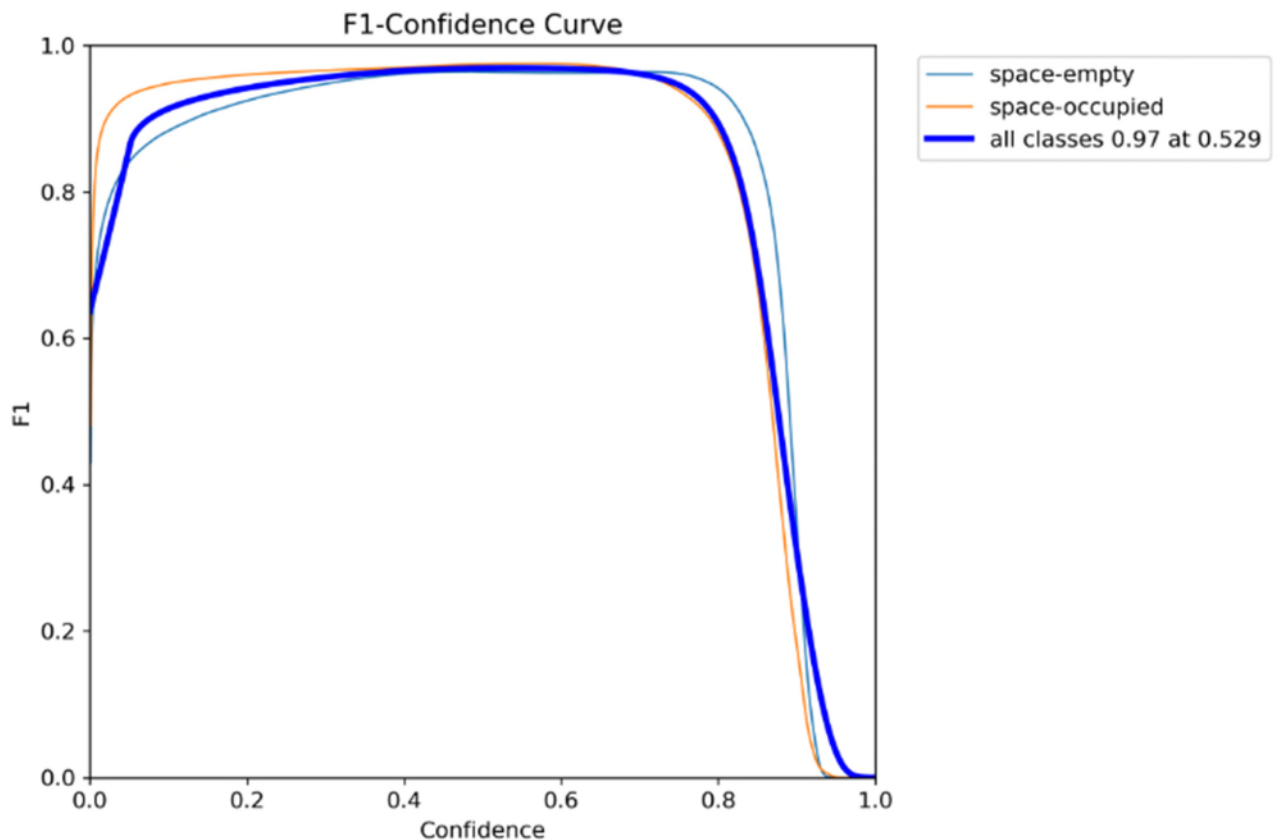


Рисунок 3.7 – Графік метрики F1-score

Графіки, побудовані на основі даних з файлу results.csv, а також зображення F1_curve.png, P_curve.png, R_curve.png і results.png дають візуальне підтвердження успішного навчання. На всіх графіках спостерігається стабільне зростання значень метрик без різких стрибків або падінь, що свідчить про ефективне навчання та правильну вибірку гіперпараметрів.

Загалом, результати аналізу метрик та графіків демонструють, що модель, яка була донавчена на 70% навчальної вибірки, досягла високої точності в задачі бінарної класифікації стану паркувального місця. Усі ключові метрики – точність, чутливість, F1-score – знаходяться на рівні 96–97%, що є винятковим результатом.

Графіки свідчать про стабільність навчання, відсутність перенавчання, і високу узагальнювальну здатність моделі.

Таким чином, модель може бути впроваджена у практичні рішення, що забезпечують автоматизоване управління паркінгом, моніторинг зайнятості та інформування користувачів у режимі реального часу, гарантуючи надійність і ефективність на рівні промислових стандартів.

3.2.2 Результати тестування моделі YOLO на тестовій вибірці

Для перевірки ефективності розробленої системи автоматичного виявлення паркувальних місць було проведено тестування моделі на незалежній тестовій вибірці, яка становить 20% від загального обсягу підготовленого датасету. Метою даного етапу є об'єктивна оцінка здатності моделі коректно визначати стан паркувальних місць (вільне або зайняте) в умовах, наближених до реального середовища. Важливо враховувати, що модель повинна функціонувати стабільно незалежно від змін умов освітлення, кута зйомки, погодних факторів, наявності тіней, часткового перекриття автомобілів чи сторонніх об'єктів.

Тестова вибірка включала зображення з різноманітними сценаріями паркування, що дозволяє повноцінно оцінити генералізаційну здатність моделі YOLO. Для візуалізації результатів було обрано одне з репрезентативних зображень, на якому наочно продемонстровано детекцію паркувальних місць з відповідними розпізнаними класами. Результати обробки містять не лише кількісну інформацію про виявлені об'єкти, але й вивід основних метрик якості моделі, зокрема: Accuracy, Precision, Recall та F1-Score.

На рисунку 3.8 зображено велику паркувальну зону, наповнену транспортними засобами, що припарковані у декілька щільно розташованих рядів. Модель YOLO здійснила обробку цього зображення, автоматично класифікуючи кожне паркувальне місце на одну з двох категорій: вільне або зайняте.

Загалом система виявила 42 паркувальних місця, з яких 29 класифіковано як вільні, а 13 – як зайняті. Розмітка візуально точна: контури охоплюють

відповідні зони без зміщення, перекриття або помилкового позиціювання. Усі виявлені об'єкти мають чітку геометрію, яка відповідає реальним розмірам паркувальних місць, що свідчить про якісну обробку просторової інформації.

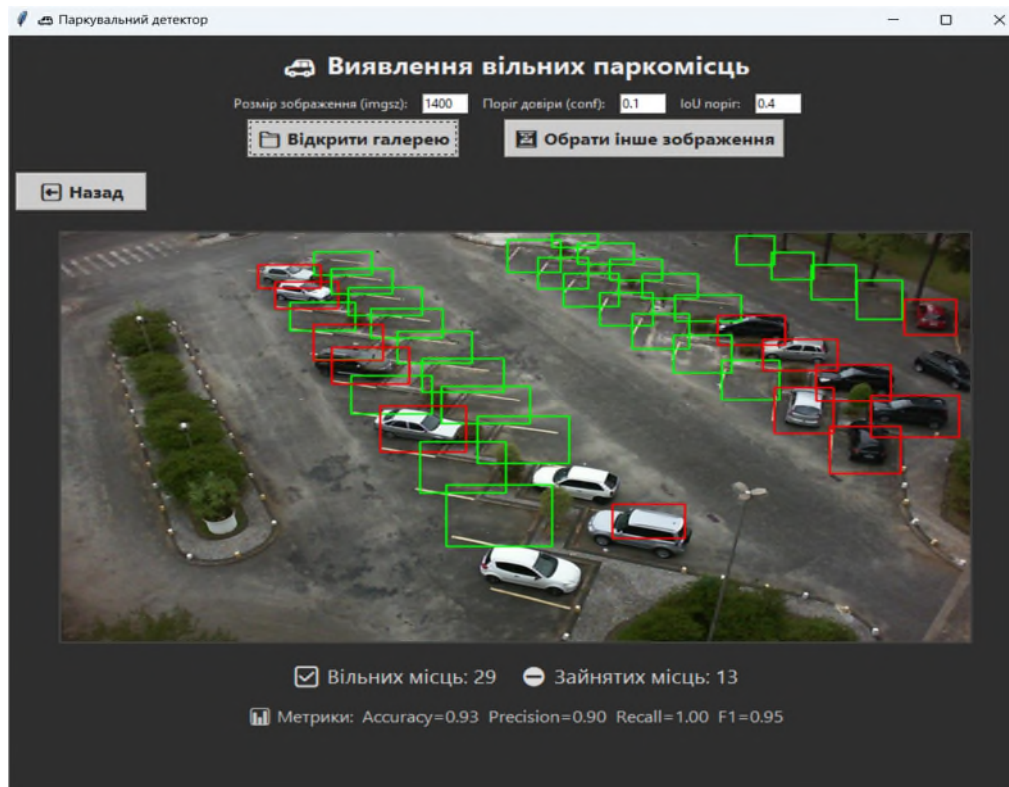


Рисунок 3.8 – Зображення паркувальних місць

Особливої уваги заслуговує точність детекції у складних умовах – наприклад, під кутом, з великою кількістю щільно припаркованих автомобілів. Видно, що модель не тільки здатна виявити вільні ділянки навіть у нижньому ряду з частковим затемненням, але й коректно ідентифікує зайняті місця попри присутність сусідніх машин, що майже перекривають одна одну. Така поведінка свідчить про здатність моделі опрацьовувати зображення з високою кількістю об'єктів, не втрачаючи при цьому точності.

Під зображенням, окрім числового підсумку про кількість вільних та зайнятих місць, виведено значення ключових метрик класифікації, що дозволяють кількісно оцінити якість роботи моделі.

Accuracy = 0.93 – 93% усіх паркомісць були класифіковані правильно.

Precision = 0.90 – 90% місць, визначених як певний клас (вільне/зайняте), дійсно відповідали цьому класу.

Recall = 1.00 – модель виявила 100% усіх реальних прикладів кожного класу (нічого не пропущено).

F1-Score = 0.95 – збалансований показник точності та повноти, що вказує на надійну загальну ефективність.

Ці метрики свідчать про високу стабільність та ефективність моделі. Система не лише точно класифікує більшість об'єктів, але й не пропускає реальні зайняті або вільні місця, що критично для практичного застосування в умовах реального міського середовища.

Для подальшої оцінки ефективності моделі YOLO використано зображення паркінгу з низькою щільністю заповненості. На рисунку 3.9 присутні лише зайняті паркувальні місця та вільних, що створює незбалансовану вибірку для класифікації.

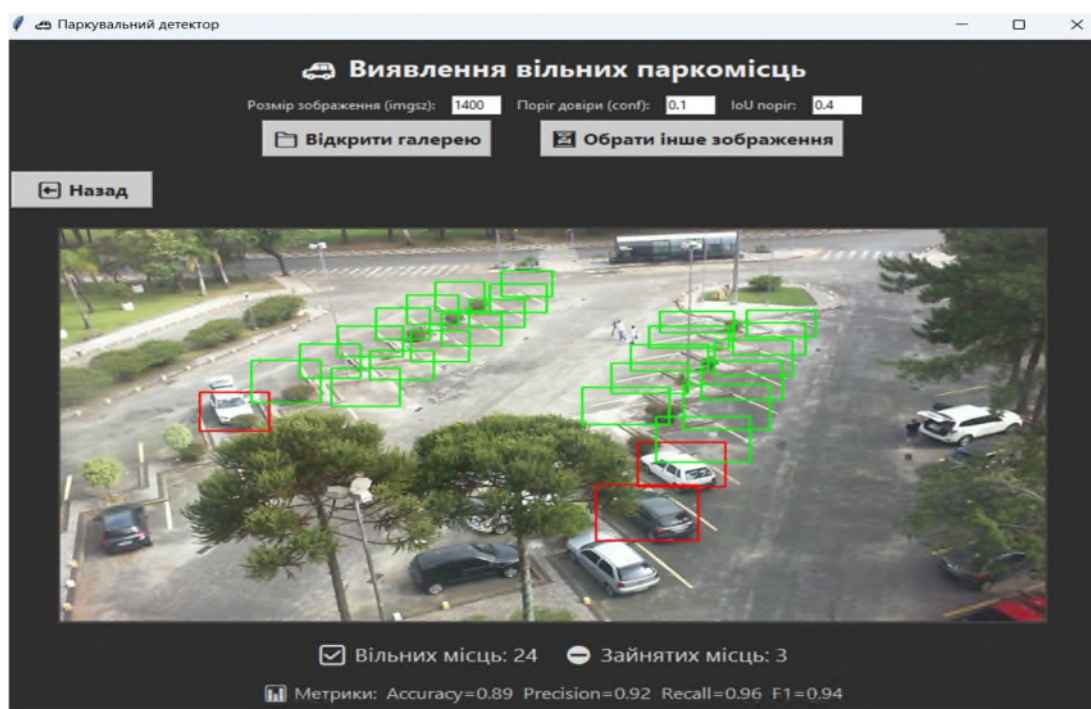


Рисунок 3.9 – Зображення паркувальних місць

Модель коректно ідентифікує зайняті місця, позначивши їх червоними рамками, тоді як решта паркувальних місць відмічені зеленим контуром як вільні.

Візуально виявлення здійснено точно – без пропусків чи хибнопозитивних спрацювань. Метрики класифікації показують: Precision = 0.92, Recall = 0.96, F1-Score = 0.94, що є наслідком великої кількості позитивних прикладів.

Загальна точність класифікації (Accuracy = 0.89) залишається високою, оскільки більшість паркувальних місць – вільні – були розпізнані правильно. Це свідчить про те, що модель добре масштабується до ситуацій з великим числом вільних місць.

На основі аналізу результатів, представлених на зображенні, можна зробити висновок про високу ефективність моделі YOLO у задачі виявлення вільних і зайнятих паркувальних місць. Отримані метрики підтверджують, що модель працює стабільно в умовах, які характерні для реального міського середовища – незалежно від зміни освітлення, щільності паркування або кута огляду. Збалансовані значення Precision, Recall і F1-Score свідчать про відсутність систематичних перекосів у сторону хибнопозитивних або хибнонегативних прогнозів.

Отже, результати оцінювання на окремих зображеннях підтвердили високу ефективність розробленої моделі як у щільних, так і в розріджених умовах паркування. У середньому по всій тестовій вибірці значення основних метрик становили: Accuracy – 0.91, Precision – 0.91, Recall – 0.98, F1-Score – 0.94. Це свідчить про стабільну роботу моделі, високу чутливість до виявлення зайнятих місць та незначну кількість хибних класифікацій.

Це дозволяє стверджувати, що розроблена метод може бути ефективно інтегрована у реальні застосунки – наприклад, для моніторингу паркінгів у режимі реального часу, побудови інтелектуальних навігаційних сервісів або автоматичного ведення статистики заповненості. Таким чином, проведене тестування підтверджує практичну доцільність та технічну готовність моделі до розгортання в продуктивному середовищі.

3.3 Висновки до розділу 3

Наведено практичну реалізацію методу визначення вільних паркувальних місць за допомогою нейромережових технологій. Розглянуто засоби, які були використані для розробки програмного застосунку, зокрема середовище розробки, мови програмування, бібліотеки та фреймворки. Обґрунтовано вибір технологій, які забезпечили ефективну інтеграцію моделі YOLO з графічним інтерфейсом користувача та можливістю обробки вхідних зображень у зручному форматі.

Докладно описано структуру програмного забезпечення, модулі та взаємозв'язки між ними. Реалізовано механізми завантаження зображень, виконання інференсу за допомогою нейромережової моделі, візуалізації виявлених паркомісць та класифікації їх як вільних або зайнятих.

У ході тестування було отримано результати донавчання моделі YOLO на підготовленій навчальній вибірці. У процесі тестування на незалежній вибірці модель продемонструвала високі середні показники якості: Accuracy – 0.91, Precision – 0.91, Recall – 0.98, F1-Score – 0.94, що підтверджує її готовність до практичного використання в системах моніторингу паркування.

Навчання показало стабільне зниження функції втрат і поступове зростання точності на валідаційних даних. Модель продемонструвала здатність ідентифікувати паркомісця з високою точністю на даних, близьких до навчальних.

Під час тестування на окремій тестовій вибірці модель підтвердила свою узагальнювальну здатність. Отримані значення метрик – таких як precision, recall, accuracy та f1-score – свідчать про високу якість виявлення і класифікації паркомісць. Таким чином, реалізовано повноцінний метод, що дозволяє визначати вільні місця на відкритих паркуваннях.

Загальні висновки

У результаті виконання кваліфікаційної роботи бакалавра на тему «Метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами» досягнуто поставлену мету – розроблено метод та інформаційну систему, яка автоматично визначає вільні й зайняті паркувальні місця за зображеннями з відкритих майданчиків. Основою розробки є методи комп'ютерного зору із застосуванням нейромережі YOLO (You Only Look Once), яка забезпечує високу точність і швидкодію в задачах детекції.

Для реалізації було використано Python, OpenCV, PyTorch, платформу Ultralytics та бібліотеку customtkinter для створення графічного інтерфейсу. Проведено навчання, валідацію та тестування моделі. Результатом є застосунок, який виконує: завантаження зображень або відео, детекцію паркувальних місць, візуалізацію результатів, підрахунок кількості вільних/зайнятих місць, розрахунок метрик (Accuracy, Precision, Recall, F1).

Система продемонструвала відповідність поставленим завданням. Модель показала точність детекції (Precision) понад 90%, а середнє значення F1-міри на тестовій вибірці становило понад 88%, що свідчить про ефективність запропонованого методу. Модель стійка до змін освітлення, погодних умов і часткового перекриття об'єктів.

Розроблений метод має перспективу впровадження у системи «розумного міста», моніторинг інфраструктури та навігацію. Систему можна вдосконалити шляхом інтеграції GPS-навігації, обробки відеопотоку в реальному часі та активного навчання для адаптації до нових умов.

Перелік посилань

1. У Львові відкривають ще 4 паркувальні майданчики, 2023. Львівська міська рада. URL: <https://city-adm.lviv.ua/news/city/transport/296270-u-lvovi-vidkryiut-shche-4-parkuvalni-maidanchyky>.
2. У Львові відкривають ще 4 паркувальні майданчики. Львівська міська рада. URL: <https://city-adm.lviv.ua/news/city/transport/296270-u-lvovi-vidkryiut-shche-4-parkuvalni-maidanchyky>.
3. Лобанчикова Наталія. ТЕХНОЛОГІЇ EDGE COMPUTING ПРИ ПОБУДОВІ ІоТ СИСТЕМИ ОХОРОНИ ПЕРИМЕТРУ. Житомир. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2021/05/51-4.pdf>.
4. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КІБЕРБЕЗПЕКА В УМОВАХ ВОЄННОГО ЧАСУ. Київ : Держ. торг.-екон. ун-т, 2024. 482 с. URL: <https://knute.edu.ua/file/MzEyMQ==/86f6b41f81cadefcbe5fb4d5d4ab277e.pdf>.
5. Powering connected vehicle services for millions of drivers globally. Parkopedia. URL: <https://business.parkopedia.com/home>.
6. Правила паркування транспортних засобів. LegalAid Wiki. URL: https://legallaid.wiki/index.php/Правила_паркування_транспортних_засобів.
7. Типи камер відеоспостереження. Ajax. URL: <https://ajax.systems.ua/blog/types-of-surveillance-cameras/>.
8. Матвієнко С. Моделі нейронних мереж. itmaster. URL: <https://itmaster.biz.ua/programming/vision/neural-networks-principles.html>.
9. Відеоспостереження на робочому місці. Wikipedia. URL: https://legallaid.wiki/index.php/Відеоспостереження_на_робочому_місці.
10. Обробка зображень. Wikipedia. URL: https://uk.wikipedia.org/wiki/Обробка_зображень.
11. Аналіз зображень. help.youscan.io. 2024. URL: <https://help.youscan.io/uk/articles/3178348-аналіз-зображень-за-допомогою-візуальні-інсайти>.

12. Артюх В. Методика визначення вільного паркувального місця на основі машинного навчання. Київ, 2021. 54 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/9fd4ab3b-d969-41c4-acc3-00039726b32d/content>.

13. Гергардт С. Алгоритм детекції рухомих об'єктів у відеопотоці на основі адаптивного ковзаючого вікна. Тернопіль, 2021. 74 с. URL: https://dspace.wunu.edu.ua/bitstream/316497/43883/1/Гергардт_С.О._КІМ-21_original_24012022_084246.pdf.

14. Назаркевич М. Система розпізнаванням об'єктів на основі моделі Yolo. Science.lpnu.ua. 2023. URL: <https://science.lpnu.ua/uk/ujit/vsi-vypusky/tom-6-nomer-1-2024/systema-rozpiznavannyam-obyektiv-na-osnovi-modeli-yolo>.

15. Custom Object Detection with YOLO – Parking Lot Occupancy Model. Clairvoyant.ai. URL: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQT1R-mZhUx-EjZrbos-9xdRf4FcNj2f4ImLg&am;s>.

16. Рибалко С. Використання OpenCV. Evergreen. URL: <https://evergreens.com.ua/ua/articles/open-cv-face-recognition.html>.

17. Артюх В. Методика визначення вільного паркувального місця на основі машинного навчання. Київ, 2021. 54 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/9fd4ab3b-d969-41c4-acc3-00039726b32d/content>.

18. Олексів Н. Метод ідентифікації машин на основі YOLO. science.lpnu.ua. 2024. URL: <https://science.lpnu.ua/uk/sisn/vsi-vypusky/vypusk-15-2024/metod-identyfikaciyi-boyovuyh-mashyn-na-osnovi-yolo>.

19. Оптимізація ефективності: впровадження системи паркування. Smart Parking International. URL: <https://www.smartparkingintl.com/uk/a-news-optimize-efficiency-lpr-parking-system-implementations>.

20. Методи обробки зображень. nmetau.edu.ua. URL: https://nmetau.edu.ua/file/07_7.5_lbr_gr_rbr_.pdf.

21. Smart Parking Detection System with YOLOv8 and OpenCV. GitHub. URL: <https://github.com/Arpitpatel1706/car-parking-slot-occupancy-detection-using-YOLOv8---openCV>.
22. Smart Parking Detection System with YOLOv8 and OpenCV (скріншот роботи проекту). GitHub. URL: <https://github.com/Arpitpatel1706/car-parking-slot-occupancy-detection-using-YOLOv8---openCV>.
23. da Luz G. Smart Parking System with Pixel-wise ROI. 2024. URL: https://www.researchgate.net/publication/386419334_Smart_Parking_with_Pixel-Wise_ROI_Selection_for_Vehicle_Detection_Using_YOLOv8_YOLOv9_YOLOv10_and_YOLOv11.
24. Smart Parking System with Pixel-wise ROI. Фото роботи проекту. URL: https://www.researchgate.net/publication/386419334/figure/fig4/AS:11431281294849958@1733328500548/Sample-of-the-post-processing-approach_Q320.jpg
25. SMART PARKING SYSTEM FOR LICENSE PLATE RECOGNITION BASED ON YOLO. science.lpnu.ua. URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2024/dec/37417/2515.pdf>.
26. Ваколюк В. Інформаційна технологія розпізнавання об'єктів. Вінниця, 2023. 132 с. URL: <https://iq.vntu.edu.ua/repository/getfile.php/7952.pdf>.
27. Архітектура YOLO 12. LearnOpenCV. URL: <https://learnopencv.com/yolov12/>.
28. Нейромережева модель. ultralytics.com. URL: <https://docs.ultralytics.com/ru/models/yolo12/>.
29. Parking Space Detection & Classification Dataset. kaggle.com. URL: <https://www.kaggle.com/datasets/trainingdatapro/parking-space-detection-dataset>.
30. Parking Space Detection & Classification Dataset. kaggle.com. URL: <https://www.googleapis.com/download/storage/v1/b/kaggle-user-content/o/inbox/12421376/cfcbb4ab9835f1d0438660e9e716edc7/MacBook%20Air%20-%201.png?generation=1691494918451033&alt=media>.
31. Bounding box annotation guide. HiTechBPO. URL: <https://www.hitechbpo.com/blog/bounding-box-annotation-guide.php>.

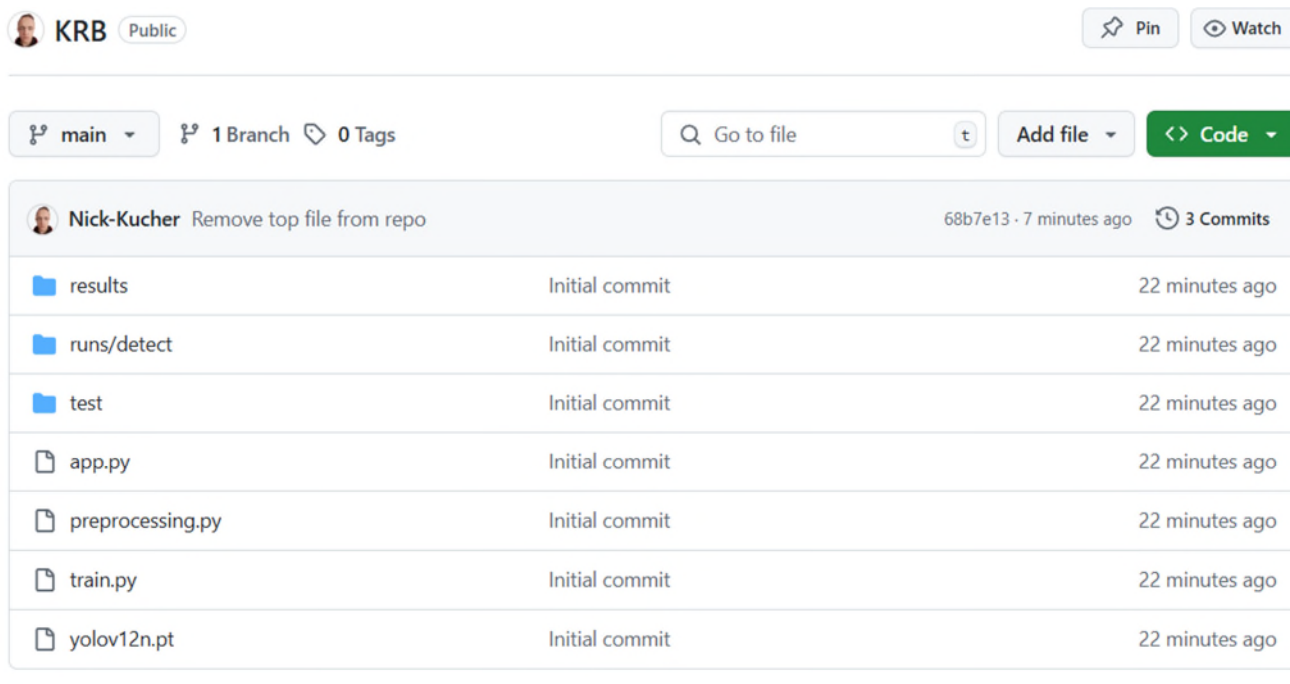
32. Коросташовець А. Відмінності та спільне міжточністю та прецизійністю вимірювань. worksection. 2023. URL: <https://worksection.com/ua/blog/accuracy-vs-precision.html>.
33. Коросташовець А. Відмінності та спільне міжточністю та прецизійністю вимірювань. worksection. 2023. URL: <https://worksection.com/ua/blog/accuracy-vs-precision.html>.
34. Recall. ultralytics.com. URL: <https://www.ultralytics.com/glossary/recall>.
35. Kundu R. F1 Score in Machine Learning. v7labs. 2022. URL: <https://www.v7labs.com/blog/f1-score-guide>.
36. Confusion Matrix. gstatic.com. URL: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQkGqoqhSnrts7erQssfJjGBf5yI5BxWbzI4w&s>.
37. Torovets T. ROC-криві. Оглядова стаття. dou.ua. 2021. URL: <https://dou.ua/forums/topic/33858/>.
38. Що таке мова програмування Python?. freehost.com.ua. URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-jazik-programmirovanija-python/>.
39. PyTorch. noscuadih. URL: https://www.noscuadih.org/?page_id=605.
40. Розгортання моделей Ultralytics. evo.net.ua. 2025. URL: <https://evo.net.ua/deploying-ultralytics-yolo-models-on-raspberry-pi-devices/>.
41. Ultralytics. YOLO. URL: <https://yolov8.com/>.
42. Матвієнко С. Вступ до OpenCV. Комп'ютерний зір. itmaster.biz.ua/. 2025. URL: <https://itmaster.biz.ua/programming/vision/opencv.html>.
43. Python Tkinter: основи для початківців. foxminded.ua. 2024. URL: <https://foxminded.ua/tkinter-python-yak-vykorystovuvaty/>.

ДОДАТКИ

Додаток А

Програмний код

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



The screenshot shows the GitHub interface for the repository 'KRB' by user 'Nick-Kucher'. The repository is public and has 1 branch and 0 tags. The commit history shows a recent commit by 'Nick-Kucher' titled 'Remove top file from repo' at 68b7e13, 7 minutes ago, with 3 commits. The file list includes folders 'results', 'runs/detect', and 'test', and files 'app.py', 'preprocessing.py', 'train.py', and 'yolov12n.pt', all from the initial commit 22 minutes ago.

File/Folder	Commit	Time
results	Initial commit	22 minutes ago
runs/detect	Initial commit	22 minutes ago
test	Initial commit	22 minutes ago
app.py	Initial commit	22 minutes ago
preprocessing.py	Initial commit	22 minutes ago
train.py	Initial commit	22 minutes ago
yolov12n.pt	Initial commit	22 minutes ago

Папка `results` містить збережені результати роботи моделі, зокрема оброблені зображення або метрики. Папка `runs/detect` — це стандартна структура, яка створюється бібліотекою YOLO під час виконання детекції об'єктів; вона містить результати зображень із нанесеними рамками виявлених об'єктів. Папка `test` включає основні скрипти для функціонування проєкту. Файл `app.py` є головним застосунком, який запускає застосунок, де можна завантажити зображення, запустити детекцію та переглянути результати. Файл `preprocessing.py` відповідає за підготовку зображень до подачі в модель: зміна розміру, нормалізація, можливо, видалення шуму тощо. Файл `train.py` використовується для навчання моделі, швидше за все, за допомогою бібліотеки YOLO, і містить логіку завантаження датасету, налаштування гіперпараметрів і запуску тренування.

Додаток Б

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

МЕТОД ВИЗНАЧЕННЯ ВІЛЬНИХ МІСЦЬ НА ВІДКРИТИХ ПАРКУВАННЯХ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖЕВИХ ЗАСОБІВ

ВИКОНАВ : СТУДЕНТ ГРУПИ КН-21-1 КУЧЕР МИКОЛА

КЕРІВНИК: ДОЦЕНТ КАФЕДРИ КН РУСЛАН БАГРІЙ

1

МЕТА ТА ЗАВДАННЯ

МЕТОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА Є ПІДВИЩЕННЯ ТОЧНОСТІ ВИЗНАЧЕННЯ ВІЛЬНИХ МІСЦЬ НА ВІДКРИТИХ ПАРКУВАННЯХ ІЗ ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ YOLOV12, ЩО ВИКОНУЄ ФУНКЦІЇ ВІЯВЛЕННЯ, КЛАСИФІКАЦІЇ ТА ВІЗУАЛІЗАЦІЇ ПАРКУВАЛЬНИХ МІСЦЬ НА ОСНОВІ ВІДЕОПОТОКУ З КАМЕР СПОСТЕРЕЖЕННЯ. ДЛЯ ДОСЯГНЕННЯ ПОСТАВЛЕНОЇ МЕТИ ПОТРІБНО ВИКОНАТИ НАСТУПНІ ЗАВДАННЯ:

- ПРОВЕСТИ АНАЛІЗ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ ТА ІСНУЮЧИХ ПІДХОДІВ ДЛЯ ВИЗНАЧЕННЯ ВІЛЬНИХ МІСЦЬ НА ВІДКРИТИХ ПАРКУВАННЯХ;
- РОЗРОБИТИ МЕТОД ВИЗНАЧЕННЯ ВІЛЬНИХ МІСЦЬ НА ВІДКРИТИХ ПАРКУВАННЯХ З ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ;
- ПІДГОТУВАТИ НАБІР ДАНИХ, ЩО МІСТИТЬ РІЗНОМАНІТНІ ЗОБРАЖЕННЯ ВІДКРИТИХ ПАРКУВАННЯХ ДЛЯ ДОНАВЧАННЯ МОДЕЛІ;
- ПРОВЕСТИ ВАЛІДАЦІЮ ЗАПРОПОНОВАНОГО МЕТОДУ ЗА ВІДОМИМИ СТАТИСТИЧНИМИ ПОКАЗНИКАМИ.

2

ОПИС ОБРАНОГО ДАТАСЕТУ

Для реалізації завдання автоматичного визначення зайнятості паркувальних місць було використано датасет PKLot.v1-raw, що містить реальні зображення відкритих парковок, отримані з камер відеоспостереження за різних погодних умов та освітлення. Знімки охоплюють різні кути огляду, моделюючи реальні сценарії розміщення камер. Особливістю набору є точна ручна розмітка у форматі YOLO: кожне паркомісце позначене обмежувальним прямокутником із класом – «space-occupied» або «space-empty». Завдяки єдиному формату розмітки датасет легко інтегрується у стандартні пайплайни для навчання моделей виявлення об'єктів. Усього датасет містить 12 439 зображень, які розподілено у співвідношенні: 70% – тренувальні, 20% – валідаційні, 10% – тестові.

```

  dataset
  ├── test
  │   ├── images
  │   └── labels
  ├── train
  │   ├── images
  │   ├── labels
  │   └── labels.cache
  ├── valid
  │   ├── images
  │   └── labels
  │       └── labels.cache
  ├── data.yaml
  ├── README.dataset.txt
  └── README.roboflow.txt
  
```

3

ЗАГАЛЬНА СХЕМА МЕТОДУ ВИЗНАЧЕННЯ ВІЛЬНИХ МІСЦЬ НА ВІДКРИТИХ ПАРКУВАННЯХ НЕЙРОМЕРЕЖЕВИМИ ЗАСОБАМИ



Крок 1. Вхідні дані. Отримується зображення з камери, встановленої так, щоб охоплювати всю парковку. Зображення має бути чітким і високої якості.

Крок 2. Попередня обробка зображення. Фото готується до обробки: масштабування, очищення від шумів тощо.

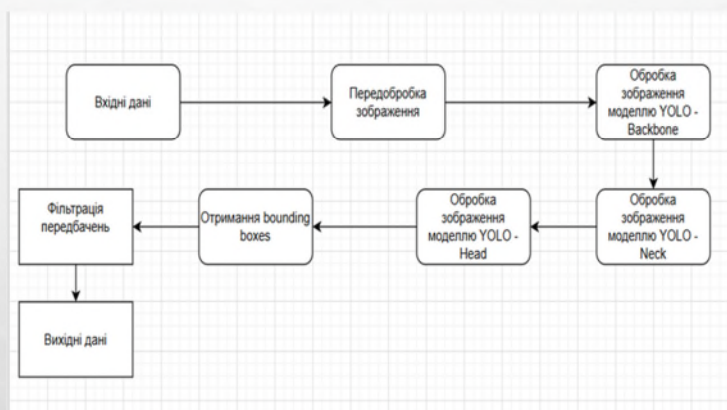
Крок 3. Детекція автомобілів. Модель YOLO виконує виявлення авто, створюючи прямокутники навколо об'єктів із зазначенням класу й рівня впевненості.

Крок 4. Аналіз зайнятості. Порівнюються координати авто з зонами паркомісць. Якщо є перекриття — місце зайняте, якщо ні — вільне

Крок 5. Вихідні дані. Результати візуалізуються: вільні місця — зелені, зайняті — червоні.

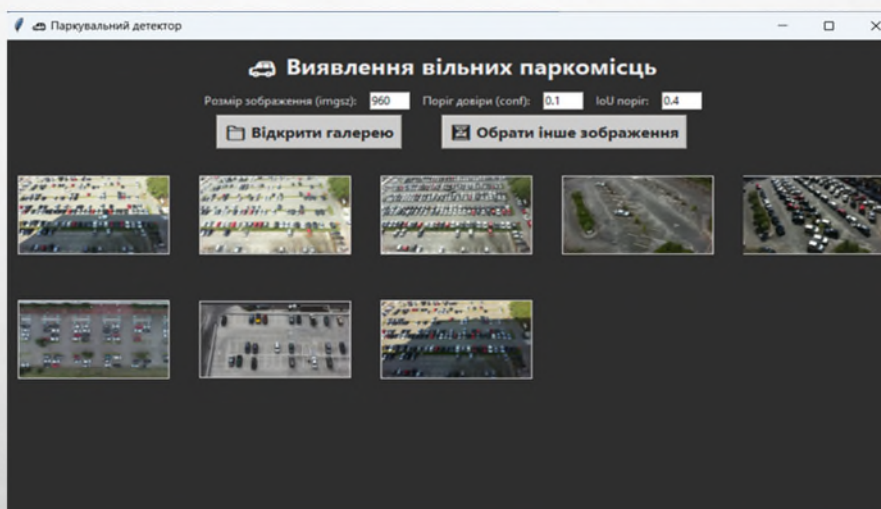
4

СХЕМА ПРОЦЕСУ ВИКОРИСТАННЯ НАВЧЕНОЇ МОДЕЛІ



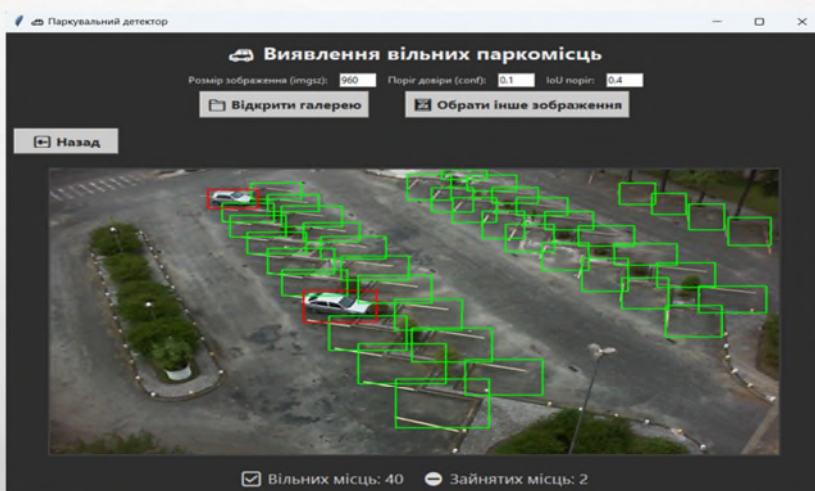
5

ГОЛОВНЕ МЕНЮ ЗАСТОСУНКУ



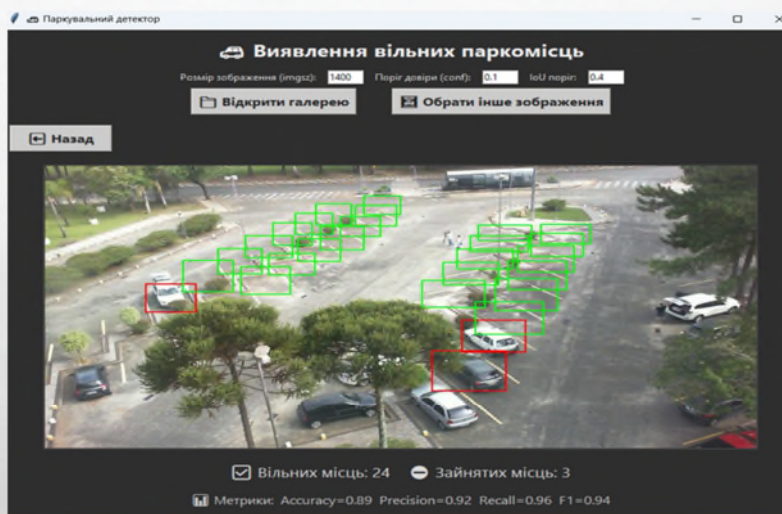
6

ПРИКЛАД ВИЗНАЧЕННЯ ВІЛЬНИХ МІСЦЬ



7

РЕЗУЛЬТАТИ МЕТРИК ПРИ ТЕСТУВАННІ



8

ВИСНОВОК

У результаті виконання кваліфікаційної роботи було реалізовано метод автоматичного визначення зайнятості паркувальних місць на основі комп'ютерного зору та глибокого навчання. Розроблений програмний застосунок виконує завантаження зображень або відео, обробку даних, детекцію автомобілів за допомогою моделі YOLO, візуалізацію результатів і розрахунок основних метричних показників (Accuracy, Precision, Recall, F1). Метод пройшов усі етапи — навчання, валідацію, тестування — та успішно продемонстрував відповідність поставленим завданням. Досягнуто високої точності: показник Precision перевищив 90%, а середнє значення F1-міри на тестовій вибірці склало понад 88%. Таким чином, мету роботи досягнуто повністю, а запропонований підхід підтвердив свою ефективність у вирішенні завдання моніторингу зайнятості паркувальних місць.

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 2.0%

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

ID: 244287 Title: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами Added in a DB: 2025-06-09 Authors: Микола КУЧЕР Heads: Руслан БАГРІЙ Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	72729	1105	3085 (4%)	52 (5%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

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

Автор: Микола КУЧЕР

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами

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

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1: 5.4%

Коефіцієнт подібності 2: 2.2%

Мікропробіли: 0

Заміна букв: 3

Інтервали: 0

Білі знаки: 20

Дата створення звіту: 2025-06-09 09:02:15.0

Після аналізу Звіту подібності констатую наступне:

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

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

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

Обґрунтування:

2025-06-09

Дата

експерт

Петровський Р. Р.

ВІСНОВКИ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод визначення вільних місць на відкритих паркуваннях за допомогою засобів

Автор студент групи КП-21-1 Микола Кучер

Освітня програма Комп'ютерні науки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки

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

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

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

Підтвердження:

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

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

- за системою Anti-Plagiarism: 2%;

- за системою StrikePlagiarism КП1: 5.43%, КП2: 0,67%.

13.06.2025

Завідувач кафедри



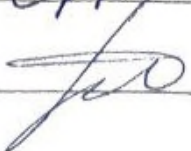
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Руслан БАГРІЙ



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента *гр. КІІ-21-1 Кучера Миколи Олександровича*

на тему *Метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами*

1. Актуальність теми

Тема є актуальною через стрімке зростання автомобільного трафіку в містах і обмежену кількість паркувальних місць, що призводить до заторів і неефективного використання простору. Застосування нейромережесвих методів, зокрема моделі YOLO, для автоматичного визначення вільних паркувальних місць забезпечує швидке й точне розв'язання цієї проблеми. Розробка сприяє підвищенню ефективності паркування та інтеграції в системи управління міською інфраструктурою.

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

Тема кваліфікаційної роботи "Метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами" відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи бакалавра. Результатом роботи є розробка методу, що базується на згортковій нейронній мережі YOLO, для обробки зображень і аналізу координат паркувальних місць. Для вирішення завдань використано методи комп'ютерного зору, підготовки даних, донавчання нейромереж і оцінки ефективності за статистичними метриками.

3. Професійні та особистісні якості бакалавра

Кучер М. О. під час виконання кваліфікаційної роботи продемонстрував достатній рівень знань у галузі комп'ютерного зору та застосування нейромережесвих технологій. Студент проявив старанність, уміння працювати з теоретичним матеріалом і практичними задачами, а також відповідальний підхід до виконання поставлених завдань.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійно, академічного плагіату не виявлено, усі запозичення оформлено з відповідними посиланнями на джерела.

5. Ступінь оволодіння методами дослідження

Студент продемонстрував достатнє розуміння теоретичних основ комп'ютерного зору та методів глибокого навчання, застосованих для класифікації об'єктів. Опанував базові підходи до аналізу зображень, підготовки даних, донавчання нейромережесих моделей і оцінки їхньої ефективності за допомогою статистичних метрик. Рівень компетентності дозволяє обґрунтувати вибір методів і провести інтерпретацію результатів, хоча окремі аспекти потребують подальшого вдосконалення.

6. Повнота та якість розкриття теми роботи

Тема роботи розкрита в достатньому обсязі: проведено аналіз сучасних нейромережесих підходів, розроблено метод визначення вільних паркувальних місць, створено програмний застосунок із графічним інтерфейсом, протестовано модель на відкритих наборах даних.

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

Матеріал викладено логічно, послідовно та з достатньою аргументованістю. Робота структурована, містить чіткий опис розділів, а мова відповідає стандартам наукових текстів.

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

Запропонований метод і програмний застосунок можуть бути використані в системах моніторингу паркінгів у міських умовах, зокрема для обробки зображень із камер відеоспостереження. Рішення має потенціал для адаптації до інших типів паркувальних майданчиків, але потребує доопрацювання для складних умов експлуатації.

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

Враховуючи достатній рівень виконання, розкриття теми та дотримання основних вимог, кваліфікаційна робота може бути допущена до захисту. Рекомендована оцінка – "добре".

Керівник _____



к.т.н., доц. Руслан Багрій



Кафедра комп'ютерних наук

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-21-1 Кучера Миколи Олександровича

за темою: Метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами

1. Актуальність обраної теми

Тема є актуальною через зростання кількості автомобілів і дефіцит паркувальних місць у міських умовах, що ускладнює пошук вільних зон для паркування. Розробка методу визначення вільних місць на відкритих паркінгах із використанням нейромережових технологій забезпечує високу точність і швидкість класифікації, оптимізуючи використання паркувальних зон. Це підвищує зручність для водіїв і сприяє розвитку "розумних" міст.

2. Повнота розкриття мети та завдань роботи

Під час виконання кваліфікаційної роботи бакалавра був реалізований метод визначення вільних місць на відкритих паркуваннях нейромережевими засобами, що відповідає меті та завданням кваліфікаційної роботи і розкриває їх повною мірою.

3. Зміст кожного розділу роботи

Кваліфікаційна робота складається з трьох розділів. Перший розділ присвячений огляду нейромережових технологій, аналізу існуючих методів обробки зображень для виявлення вільних паркомісць, а також формулює мету та завдання дослідження. Другий розділ описує проєктування методу визначення вільних місць на паркуваннях із використанням нейромережі YOLO, включаючи архітектуру, підготовку даних та метрики ефективності. Третій розділ розглядає реалізацію програмного застосунку, результати тестування моделі та аналіз точності роботи системи.

4. Оцінка розробленого методу та його практична цінність

Розроблений метод забезпечує ефективне автоматичне виявлення вільних паркувальних місць за допомогою нейромережової моделі YOLO. Це підвищує ефективність використання паркувального простору, зменшує час пошуку місця та може бути впроваджене в системи "розумного міста".

5. Якість оформлення кваліфікаційної роботи бакалавра

Записка якісно оформлена відповідно до встановлених вимог, чітко і зрозуміло написана, зі структурованою побудовою розділів та логічною послідовністю викладення матеріалу.

6. Недоліки кваліфікаційної роботи бакалавра

Рекомендовано удосконалити метод адаптації до складних умов, таких як нічна зйомка або погана погода, а також розглянути механізм автоматичного оновлення координат паркомісць для гнучкішого розгортання.

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

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

Рецензент к.т.н. доцент кафедри АКТІЧАР
Корсунька Л.О.

[Підпис]