

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Кіберфізична система визначення зайнятості паркувальних місць на основі
нейромережевого розпізнавання об'єктів та просторового аналізу

Назва теми

Рівень вищої освіти другий (магістерський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ 240101.47.01.01 ПЗ

Виконав здобувач II курсу, група КІ2М-24-1

Керівник

доктор філософії, доцент
Науковий ступінь, учене звання

Нормоконтролер

д. техн. наук, професор
Науковий ступінь, учене звання

До захисту допускаю:
завідувач кафедри КІС
01 » травня 2026 р.

дата


Підпис

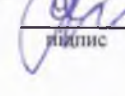
Володимир АВСІЄВИЧ
Ініціали, прізвище


Підпис

Ольга ПАВЛОВА
Ініціали, прізвище


Підпис

Сергій ЛИСЕНКО
Ініціали, прізвище


Підпис

Ольга ПАВЛОВА
Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ДРУГИЙ (МАГІСТЕРСЬКИЙ)

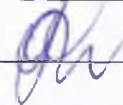
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІІС

 Ольга ПАВЛОВА

“ 12 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Авсієвичу Володимиру Руслановичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторового аналізу

Керівник проекту (роботи) Павлова О.О., д. ф., завідувач кафедри КІІС

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 12.01.2026 р. № 6

2. Термін подання здобувачем роботи на кафедру 01.05.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

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

Аналіз сучасного стану технологій моніторингу паркувального простору

Математичне моделювання процесу визначення зайнятості паркомісць

Метод просторово-часового аналізу для визначення зайнятості паркомісць

Програмна реалізація та експериментальні дослідження кіберфізичної системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 12 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	12.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	12.01.2026	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	20.01.2026	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.02.2026	виконано
5	Робота над науковою статтею	01.03.2026	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.03.2026	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2026	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2026	виконано
9	Попередній захист ДРМ	29.04.2026	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2026	


Здобувач


Підпис

Володимир АВСІЄВИЧ

Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Ольга ПАВЛОВА

Імя, ПРІЗВИЩЕ

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Кіберфізична система визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторового аналізу.

Автор роботи: Авсієвич Володимир Русланович.

Керівник роботи: Павлова Ольга Олександрівна.

Пояснювальна записка: 88 с., 27 рис., 3 табл., 3 дод., 83 джерела.

КОМП'ЮТЕРНИЙ ЗІР, КІБЕРФІЗИЧНА СИСТЕМА, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, МОНІТОРИНГ ПАРКУВАННЯ, ПРОСТОРОВО-ЧАСОВИЙ АНАЛІЗ, ОПТИМІЗАЦІЯ ОБЧИСЛЕНЬ, YOLO.

Об'єктом дослідження є процес автоматизованого візуального моніторингу завантаженості паркувального майданчика.

Предметом дослідження є методи та алгоритми просторово-часового аналізу відеоданих на основі згорткових нейронних мереж.

Метою кваліфікаційної роботи магістра є вдосконалення існуючих методів визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторово-часового аналізу відеозображень для підвищення точності моніторингу та забезпечення роботи системи в умовах жорстко обмежених обчислювальних ресурсів.

Для розв'язання поставлених задач використовувалися методи комп'ютерного зору та розпізнавання образів, методи обчислювальної геометрії для просторового аналізу, принципи теорії автоматів для часової фільтрації станів, а також методи математичного та комп'ютерного моделювання процесів.

Наукова новизна отриманих результатів:

– набув подальшого розвитку метод просторово-часового аналізу для визначення зайнятості паркомісць, який використовує масштабно-інваріантну метрику просторового перекриття IoMin у комбінації зі скінченним автоматом з

подвійним порогом, що дозволило зменшити вплив перспективних спотворень та усунути коливання станів від маневруючого транспорту;

– удосконалено архітектуру програмного забезпечення кіберфізичної системи відеомоніторингу за рахунок впровадження моделі дискретного підключення до відеопотоку та алгоритмічних оптимізацій з забезпеченням відмовостійкості, що забезпечило її безперебійну роботу на обчислювальних потужностях центрального процесора без залучення дискретних графічних прискорювачів.

Практична значимість отриманих результатів полягає у розробленій апаратно-незалежній кіберфізичній системі візуального моніторингу паркувального простору. Створений програмний комплекс здатний функціонувати в умовах обмежених ресурсів обчислювальних вузлів виключно на потужностях центрального процесора, що знижує вартість розгортання таких систем.

У першому розділі проведено комплексний аналіз сучасного стану технологій та методів візуального моніторингу паркувального простору, виявлено недоліки існуючих комерційних рішень та обґрунтовано вибір одностадійних згорткових нейронних мереж як базового інструменту локалізації об'єктів для роботи в режимі реального часу.

У другому розділі розроблено математичні моделі просторового відбору об'єктів та часової стабілізації станів для компенсації ефекту візуального паралаксу. Формалізовано концепцію системи, критерії оцінки якості класифікації та логіку роботи часового фільтра на базі скінченного автомата з подвійним порогом.

У третьому розділі удосконалено метод просторово-часового аналізу для визначення зайнятості паркомісць. Впроваджено адаптивну метрику просторового перекриття та проведено комп'ютерне моделювання, яке експериментально довело підвищення повноти та точності класифікації станів паркомісць із зменшенням кількості хибних спрацювань від динамічних перешкод.

У четвертому розділі спроектовано архітектуру та здійснено практичну програмну реалізацію кіберфізичної системи. Проведено експериментальну оцінку

та оптимізацію ресурсної ефективності програмного комплексу, що підтвердило його здатність стабільно працювати на базі центрального процесора в умовах жорстких квот.

ЗМІСТ

Скорочення та умовні позначки	5
Вступ.....	6
1 Аналіз сучасного стану технологій моніторингу паркувального простору.....	9
1.1 Аналіз існуючих апаратно-програмних комплексів розумних парковок.....	9
1.2 Огляд методів комп'ютерного зору та архітектур нейронних мереж для пошуку об'єктів	14
1.2.1 Еволюція методів розпізнавання автомобілів.....	14
1.2.2 Порівняльний аналіз архітектур двостадійних та одностадійних розпізнавачів.....	18
1.2.3. Архітектурні особливості сімейства YOLO та обґрунтування вибору моделі	22
1.3 Аналіз методів просторової фільтрації та просторового аналізу зображень	24
1.4 Висновки і постановка задачі дослідження.....	28
2 Математичне моделювання процесу визначення зайнятості паркомісць	31
2.1 Обґрунтування вибору методу та розробка загальної концепції системи ..	31
2.2 Математична модель сцени паркування, геометричних перетворень та розпізнавання об'єктів	36
2.2.1 Математична модель сцени	36
2.2.2 Математична модель фільтрації множини розпізнаних об'єктів	40
2.3 Математичне моделювання просторового відбору об'єктів на основі полігональних масок	42
2.3.1 Перевірка належності точки багатокутнику	42
2.3.2 Оцінка площі просторового перекриття.....	46
2.3.3 Формування комплексного критерію зайнятості	47
2.4 Математичне моделювання часової динаміки та фільтрації станів.....	49
2.5 Висновки	53
3 Метод просторово-часового аналізу для визначення зайнятості паркомісць..	55

3.1	Загальна структура розробленого методу просторово-часової фільтрації .	55
3.2	Метод попередньої обробки відеопотоку та просторового маскуваня	57
3.3	Метод фільтрації і семантичного відбору результатів розпізнавання	59
3.4	Метод просторового відбору транспортних засобів	60
3.5	Метод часової стабілізації станів	62
3.6	Оптимізація методу для систем з обмеженими ресурсами.....	64
3.7	Формування просторових зон інтересу та еталонних даних	66
3.8	Комп'ютерне моделювання та експериментальна перевірка розроблених методів	68
3.9	Висновки	74
4	Програмна реалізація та експериментальні дослідження кіберфізичної системи	76
4.1	Архітектура та стек технологій програмного забезпечення.....	76
4.2	Алгоритм роботи обчислювального ядра та розробка інтерфейсу користувача	79
4.3	Експериментальна оцінка продуктивності та поетапна оптимізація ресурсної ефективності системи	84
4.4	Забезпечення відмовостійкості та алгоритми обробки виняткових ситуацій	89
4.5	Висновки	90
	Висновки	92
	Перелік джерел посилань	94
	Додаток А Лістинг програмного забезпечення.....	104
	Додаток Б Публікація.....	107
	Додаток В Презентація	119

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЗНМ – згорткова нейронна мережа

IoMin – Intersection over Minimum (перетин відносно мінімуму)

YOLO – You Only Live Once (сімейство моделей нейромереж)

ROC-крива – Receiver Operating Characteristic (робоча характеристика приймача)

IoU – Intersection over Union (перетин відносно об'єднання)

ВСТУП

У сучасних умовах стрімкої урбанізації проблема ефективного управління паркувальним простором набуває критичного значення. Класичні системи моніторингу, побудовані на базі контактних апаратних датчиків, характеризуються високою вартістю розгортання та складністю обслуговування. Альтернативним і найбільш перспективним підходом є використання технологій комп'ютерного зору та згорткових нейронних мереж у складі кіберфізичних систем.

Практичне впровадження сучасних нейромережевих моделей масового сегмента стикається і з апаратною проблемою. Для забезпечення обробки відеопотоку в режимі реального часу такі системи вимагають наявності потужних і дороговартісних дискретних графічних прискорювачів. Однак реальні умови експлуатації на об'єктах інфраструктури (зокрема, на постах охорони) передбачають використання базових персональних комп'ютерів загального призначення або малопотужних периферійних пристроїв, які оснащені виключно центральними процесорами. Склалася певна суперечність між нагальними потребами у високоточному автоматизованому візуальному моніторингу та жорсткими ресурсними обмеженнями існуючої апаратної інфраструктури. Зменшити ступінь цих суперечностей через розроблення відповідних оптимізованих методів просторово-часового аналізу відеоданих має наше дослідження.

Актуальність роботи полягає в розробленні та оптимізації методу комп'ютерного зору для моніторингу паркувального простору, який дозволяє зменшити вплив візуального паралаксу та динамічних перешкод, забезпечуючи при цьому можливість повноцінного функціонування кіберфізичної системи виключно на обчислювальних потужностях центрального процесора без залучення дискретних графічних прискорювачів.

Метою кваліфікаційної роботи магістра є вдосконалення існуючих методів визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторово-часового аналізу відеозображень для

підвищення точності моніторингу та забезпечення роботи системи в умовах жорстко обмежених обчислювальних ресурсів.

Поставлена мета досягається розв'язанням таких основних завдань:

- проаналізувати сучасний стан, існуючі апаратно-програмні комплекси та методи комп'ютерного зору у сфері візуального моніторингу паркувального простору;
- розробити математичні моделі просторового відбору об'єктів та часової стабілізації станів для компенсації ефекту візуального паралаксу;
- удосконалити метод просторово-часового аналізу відеоданих шляхом впровадження адаптивної метрики просторового перекриття та алгоритмів фільтрації на базі скінченного автомата з подвійним порогом;
- спроектувати та розробити програмне забезпечення кіберфізичної системи з урахуванням архітектурної вимоги щодо виконання нейромережевого прогнозування на центральному процесорі;
- провести експериментальні дослідження для оцінки точності розпізнавання та ресурсної ефективності оптимізованої системи в реальних умовах експлуатації.

Об'єкт дослідження – процес автоматизованого візуального моніторингу завантаженості паркувального майданчика.

Предмет дослідження – методи та алгоритми просторово-часового аналізу відеоданих на основі згорткових нейронних мереж.

Наукова новизна отриманих результатів:

- набув подальшого розвитку метод просторово-часового аналізу для визначення зайнятості паркомісць, який використовує масштабно-інваріантну метрику просторового перекриття IoMin у комбінації зі скінченим автоматом з подвійним порогом, що дозволило зменшити вплив перспективних спотворень та усунути коливання станів від маневруючого транспорту;
- удосконалено архітектуру програмного забезпечення кіберфізичної системи відеомоніторингу за рахунок впровадження моделі дискретного підключення до відеопотоку та алгоритмічних оптимізацій з забезпеченням відмовостійкості, що забезпечило її безперебійну роботу на обчислювальних

потужностях центрального процесора без залучення дискретних графічних прискорювачів.

На основі проведених досліджень розроблено метод просторово-часового аналізу для визначення зайнятості паркувальних місць.

Практична значимість отриманих результатів полягає у розробленій апаратно-незалежній кіберфізичній системі візуального моніторингу паркувального простору. Створений програмний комплекс здатний функціонувати в умовах обмежених ресурсів обчислювальних вузлів виключно на потужностях центрального процесора, що знижує вартість розгортання таких систем.

Для розв'язання поставлених задач використовувалися методи комп'ютерного зору та розпізнавання образів, методи обчислювальної геометрії для просторового аналізу, принципи теорії автоматів для часової фільтрації станів, а також методи математичного та комп'ютерного моделювання процесів.

За темою кваліфікаційної роботи опубліковано одну наукову статтю [60] в електронному фаховому науковому виданні «Кібербезпека: освіта, наука, техніка» (2025. – Вип. 1 (29). – С. 536–547).

1 АНАЛІЗ СУЧАСНОГО СТАНУ ТЕХНОЛОГІЙ МОНІТОРИНГУ ПАРКУВАЛЬНОГО ПРОСТОРУ

1.1 Аналіз існуючих апаратно-програмних комплексів розумних парковок

Глобальний ринок технологій розумного паркування демонструє стійку тенденцію до зростання. За оцінками експертів, його обсяг збільшиться з 4,4 млрд доларів США у 2020 році до майже 12 млрд доларів США до 2030 року [1]. Це зростання супроводжується еволюцією архітектурних підходів до побудови кіберфізичних систем паркування. На сьогоднішній день існуючі на ринку України та світу рішення можна класифікувати за трьома основними категоріями: апаратно-орієнтовані екосистеми, хмарні рішення та системи на основі граничних обчислень.

Апаратно-орієнтовані екосистеми, історично відомі як PARCS (Parking Access & Revenue Control Systems), є стандартом для об'єктів із контрольованим доступом, таких як аеропорти, торгові центри та закриті офісні паркінги. Архітектура таких комплексів будується за топологією hub-and-spoke, де численні периферійні пристрої жорстко прив'язані до локального контролера або сервера [2].

На фізичному рівні такі системи включають електромеханічні шлагбауми, в'їзні/виїзні стійки з видачею квитків або зчитувачами RFID-карт, платіжні термінали (рис. 1.1) та індукційні петлі, вмонтовані в дорожнє покриття для розпізнавання присутності автомобіля [3]. Обмін даними між периферією та сервером зазвичай здійснюється через локальні дротові інтерфейси (Ethernet, RS-485 або промислові шини fieldbus), які забезпечують високу відмовостійкість і незалежність від зовнішнього інтернет-з'єднання.

Логіка керування зосереджена на локальному сервері, що розміщується безпосередньо на об'єкті. Типовим прикладом є рішення української компанії SEA (SEA Company), де серверна частина базується на базі даних MySQL і виконує функції розрахунку тарифів, валідації доступу та керування шлагбаумами [3]. Світові лідери, такі як SKIDATA [4] (рішення Mobility Suite)

та ПІВА Parking [5], також використовують цей підхід, де локальний сервер виступає ядром системи, забезпечуючи обробку транзакцій у реальному часі.



Рисунок 1.1 – Паркувальний термінал SEA [3]

Ключовими недоліками апаратно-орієнтованих систем є високі капітальні витрати через необхідність прокладання комунікацій та будівельних робіт, а також проблема прив'язки до виробника – пропрієтарні протоколи унеможливають просту інтеграцію обладнання сторонніх виробників без участі розробника системи [4]. Такі системи складно масштабувати в умовах вуличної інфраструктури, оскільки встановлення шлагбаумів та терміналів на кожному паркомісці є економічно недоцільним.

З розвитком Інтернету речей виник альтернативний підхід, у якому центр прийняття рішень перенесено у хмару, а фізична інфраструктура мінімізована або замінена віртуальними ідентифікаторами. Цей тип архітектури фокусується на гнучкості та інтеграції з мобільними застосунками.

Архітектура таких рішень зазвичай реалізована у двох варіантах. Перший – це чисті програмні рішення, де роль фізичного рівня виконує смартфон користувача. Яскравим прикладом є муніципальна система «Київ Цифровий» (раніше Kyiv Smart City), де водії сплачують за паркування через застосунок, а інспектори перевіряють оплату, звертаючись до хмарного реєстру [6]. У таких

системах відсутнє спеціалізоване обладнання на кшталт шлагбаумів, а фізична присутність обмежується дорожніми знаками з QR-кодами [7].

Другий варіант передбачає використання енергоефективних IoT-сенсорів. Прикладом є пілотний проєкт Lviv Startup School (Parking Cloud) у Львові, де було розгорнуто датчики Nwave (рис. 1.2) [8]. Ці пристрої, вмонтовані в асфальт, використовують комбінацію магнітометрів та радарів для розпізнавання авто [9] і передають дані через мережі LPWAN (LoRaWAN, NB-IoT) на хмарну платформу. Хмарний рівень відповідає за агрегацію даних, керування станами паркомісць, тарифікацію та надання API для сторонніх сервісів навігації чи оплати (наприклад, ParkMobile, PayByPhone) [10, 11].



Рисунок 1.2 – Датчик Nwave

Попри гнучкість і низьку вартість розгортання, такі системи мають суттєві обмеження. Вони критично залежать від стабільності каналів зв'язку. Втрата з'єднання з хмарою унеможливорює перевірку оплати або оновлення статусу зайнятості. Крім того, ґрунтові магнітні сенсори можуть давати хибні спрацювання через електромагнітні перешкоди або погодні умови [12], а питання конфіденційності даних (відповідність GDPR) стає критичним при централізованому зборі інформації про переміщення користувачів.

Третій, найбільш технологічний напрямок, передбачає перенесення обчислювальних потужностей на кінцеві пристрої за допомогою камер з вбудованими алгоритмами штучного інтелекту. Цей підхід є найбільш релевантним для завдань, що вимагають високої точності розпізнавання та автоматизації без участі людини.

Архітектура таких систем базується на використанні розумних камер як основних датчиків. Лідерами ринку в цьому сегменті є компанії Hikvision (рис. 1.3) [13], Parklio [14], Autlo [15], а також українські розробки, такі як модуль SEA ANPR [16]. Фізичний рівень включає АРНЗ-камери (автоматичне розпізнавання номерних знаків) для розпізнавання номерних знаків на в'їздах/виїздах та оглядові камери для навігації, які можуть моніторити декілька паркомісць одночасно (наприклад, до 6 місць однією камерою у рішеннях KEYTOP) [17].



Рисунок 1.3 – камера HikVision DS-2CD4626FWD-IZS(H) з ANPR

Ключовою особливістю є те, що аналіз відеопотоку відбувається локально – безпосередньо на камері або на підключеному мережевому відеореєстраторі [18]. Камера самостійно виконує знаходження об'єкта, розпізнавання символів, порівняння з локальними білими та чорними списками і навіть може напряду керувати шлагбаумом через релейні виходи (RS-485/IO), не звертаючись до центрального сервера.

Така децентралізація дозволяє значно знизити навантаження на мережу передачі даних і забезпечити миттєву реакцію системи. Наприклад, у рішенні Parklio [14] або пілотному проєкті в Києві з використанням технологій ТІВА/Afcon система автоматично ідентифікує транспортний засіб, перевіряє наявність абонементу і відкриває проїзд за долі секунди [19]. Окрім простої фіксації зайнятості, рішення цього типу дозволяють реалізувати складні сценарії: пошук конкретного авто на паркінгу, виявлення порушень правил паркування та збір аналітики поведінки водіїв.

Проведений аналіз ринкових рішень дозволяє виділити чітку тенденцію розвитку кіберфізичних систем паркування – поступовий перехід від апаратно-орієнтованих комплексів до гнучких інтелектуальних систем на базі

комп'ютерного зору. Однак попри технологічний прогрес, пряма реалізація готових комерційних рішень для розумних парковок стикається з низкою суттєвих обмежень, про які було згадано вище. Є 2 ключових групи проблем:

1. Фрагментація стандартів обміну даними та проблема прив'язки до постачальника. Критичною проблемою галузі залишається відсутність єдиного стандарту опису та передачі даних паркування. Аналіз архітектур показує, що різні класи систем використовують несумісні комунікаційні стеки:

- чисто апаратні системи спираються на закриті промислові протоколи через інтерфейси RS-485 або Ethernet;
- хмарні рішення використовують бездротові протоколи LoRaWAN/NB-IoT та прикладні протоколи MQTT/HTTPS;
- розумні камери часто інтегруються виключно через пропрієтарні API в межах екосистеми одного виробника (наприклад, HikCentral або Parklio PMS).

Таким чином, для об'єднання обладнання різних виробників (наприклад, камер Hikvision та бар'єрів SEA) необхідна розробка складного проміжного ПЗ. Крім цього, закритість архітектури не дозволяє отримати доступ до метрик роботи систем, що унеможлиблює об'єктивну оцінку їх ефективності без проведення власних тестів.

Слід також зазначити, що повна заміна існуючої інфраструктури спеціалізованими камерами або монтаж ґрунтових датчиків вимагає значних капіталовкладень, що часто є економічно недоцільним для невеликих або вже існуючих паркувальних майданчиків. Альтернативним і найбільш економічно вигідним підходом є використання відеопотоку (за протоколом RTSP) зі звичайних оглядових IP-камер, які зазвичай вже встановлені на об'єктах. У такому випадку логіка розпізнавання переноситься на локальний універсальний обчислювальний пристрій – наприклад, на стандартний персональний комп'ютер на контрольно-пропускному пункті. Це дозволяє перетворити звичайну систему відеоспостереження на інтелектуальну за мінімальних фінансових витрат та без суттєвої модифікації існуючої інфраструктури.

2. Обмеження масштабування та просторового аналізу. Комерційні системи відеовізуальної навігації, такі як KEYTOP або Hikvision, мають жорсткі

обмеження зони покриття – зазвичай одна камера обслуговує до 6 паркомісць. Для великих вуличних парковок це призводить до надмірних капіталовкладень. Більше того, ці алгоритми часто не враховують проблему перекриття об'єктів у складних ракурсах, яка є критичною при спробі моніторингу широкої зони однією оглядовою камерою.

Огляд лише ринкових рішень не дає повного розуміння усієї ситуації галузі розумних парковок, тому необхідно змістити фокус дослідження з системного рівня на алгоритмічний, а саме проаналізувати методи, які дозволяють обробляти відеопотік незалежно від апаратних складових та без прив'язки до пропрієтарних засобів.

1.2 Огляд методів комп'ютерного зору та архітектур нейронних мереж для пошуку об'єктів

1.2.1 Еволюція методів розпізнавання автомобілів

Історично розвиток систем комп'ютерного зору для розпізнавання транспортних засобів умовно ділиться на дві фундаментальні епохи: період домінування методів, що базувалися на ручному конструюванні ознак, та еру глибокого навчання, яка розпочалася приблизно у другому десятиріччі XXI століття.

Як зазначається у комплексному огляді [20], ранні системи комп'ютерного зору, розробка яких активно проводилася, починаючи з 1995 року, базувалися на класичних підходах обробки зображень та машинного навчання. Згідно з дослідженням [21], до появи глибоких нейронних мереж стандартний алгоритм обробки складався з трьох послідовних етапів: вибір регіону інтересу, вилучення ознак та класифікація.

Основною характеристикою цього періоду була необхідність ручного конструювання ознак. В огляді алгоритмів розпізнавання для автономного водіння [22] зауважують, що на цьому етапі як ознаки математично описували візуальні характеристики автомобіля, такі як краї, текстури, симетрія або колір, щоб відрізнити його від фону .

Серед найбільш поширених методів виділення ознак, які використовувалися у поєднанні з класифікаторами, виділяють три наступні методи:

1. Гістограма напрямлених градієнтів. Цей метод, вперше докладно описаний у фундаментальній праці [23], базується на аналізі розподілу напрямків градієнтів (змін інтенсивності пікселів) для опису форми та контурів об'єкта. Як підкреслюють автори сучасних оглядів [22], ГНГ показала свою ефективність у задачах виявлення пішоходів і згодом була адаптована для розпізнавання транспортних засобів, дозволяючи фіксувати структурну інформацію про об'єкт.

2. Каскади Хаара. Принцип їх застосування для виявлення ознак детально розкрито в роботі [24]. Метод використовує прямокутні примітиви (ознаки Хаара) для розрахунку різниці інтенсивностей між суміжними областями зображення. Згідно з оглядом методів глибокого навчання [25], це дозволяє виявляти прості геометричні структури та симетрію, притаманну автомобілям.

3. Локальні двійкові шаблони, дієвість яких була доведена, зокрема, у дослідженні [26]. Використовуються для опису локальних текстурних особливостей поверхні об'єкта, що дозволяє класифікувати області зображення на основі їх текстурної однорідності.

Отримані вектори ознак передавалися на вхід класифікаторів, серед яких домінуючу роль відігравали метод опорних векторів та алгоритм AdaBoost, що підтверджується результатами досліджень [27] та [28]. Крім того, у порівняльному аналізі [29] вказується, що у найпростіших реалізаціях, особливо на етапі 1995-2000 років, системи покладалися на методи віднімання фону та виявлення країв для сегментації рухомих об'єктів .

Незважаючи на певну ефективність у контрольованих умовах, традиційні методи демонстрували суттєві недоліки при застосуванні у реальних сценаріях вуличного моніторингу та парковки. Виділяють наступні фундаментальні проблеми евристичних підходів:

1. Проблема перекриття. Одним із найкритичніших недоліків є нездатність цих алгоритмів коректно розділяти об'єкти, що перекривають один одного. Методи, засновані на відніманні фону та аналізі руху, часто

ідентифікують групу транспортних засобів, що рухаються щільним потоком або перекривають один одного на парковці, як один великий об'єкт. Така поведінка алгоритмів неодмінно призводила до значних помилок у підрахунку кількості об'єктів [20]. Приклад цієї проблеми зображено на рис. 1.4.

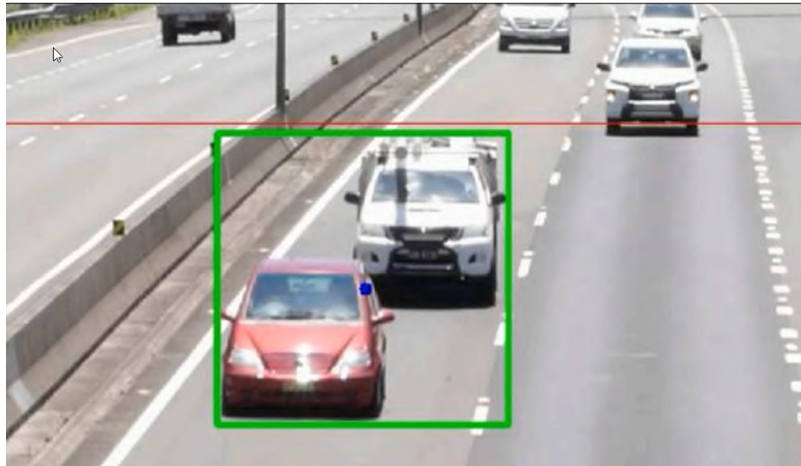


Рисунок 1.4 – Проблема перекриття [20]

2. Чутливість до освітлення та погодних умов. Точність традиційних методів сильно залежить від умов освітлення. В умовах низького освітлення (сутінки, ніч) або за наявності тіней ефективність алгоритмів різко знижується через появу шумів, що робить процес віднімання фону неефективним. Крім того, складні погодні умови, такі як дощ або туман, створюють візуальні артефакти, які евристичні алгоритми не можуть відфільтрувати. Причина цього полягає в тому, що, як доведено у порівняльному дослідженні [30] такі алгоритми не мають здатності до узагальнення даних.

3. Геометрична неадаптивність. Ранні методи покладаються на фіксовані параметри, такі як розмір об'єкта або його співвідношення сторін. Однак при зміні кута огляду камери або при наближенні транспортного засобу його візуальні характеристики змінюються, що призводить до збоїв у роботі алгоритмів, які не можуть адаптуватися до перспективних спотворень, на що також акцентується увага у роботі [31].

4. Залежність від ручного налаштування. У дослідженні [32] згадується, що традиційні рішення вимагають складного налаштування

параметрів для кожної конкретної сцени й не здатні адаптуватися до динамічних змін середовища без втручання людини.

Переломним моментом в еволюції методів комп'ютерного зору вважається період після 2010 року, коли розвиток обчислювальних потужностей та поява великих наборів даних дали можливість ефективно використовувати згорткові нейронні мережі [33]. Справжній прорив відбувся з появою архітектури AlexNet у 2012 році та подальшим розвитком розпізнавачів, таких як модель YOLO (You Only Look Once), вперше представлена в статті [34], та модель Faster R-CNN [35].

Фундаментальна відмінність підходу глибокого навчання від традиційних методів полягає у відмові від ручного конструювання ознак. У праці [36] підкреслюють, що замість використання попередньо визначених евристик (як у ГНГ або каскадах Хаара), ЗНМ автоматично навчаються виділяти ієрархічні ознаки безпосередньо з необроблених піксельних даних у процесі тренування. Перші шари мережі виділяють прості примітиви (лінії, кути), тоді як глибші шари формують складні абстракції (колеса, форми кузова), що дозволяє системі формувати більш повне та стійке уявлення про об'єкт.

Методи глибокого навчання у задачах виявлення транспортних засобів мають багато переваг. По-перше, вони надійні та стійкі до перешкод. Глибокі нейронні мережі демонструють значно вищу стійкість до складних умов навколишнього середовища. Розробники YOLO наголошують, що глибокі мережі здатні ефективно працювати за наявності часткового перекриття об'єктів, змін освітлення та в різних погодних умовах, де традиційні методи зазнають невдачі [37]. Здатність ЗНМ до узагальнення дозволяє їм ігнорувати несуттєві варіації фону та шумів.

По-друге, вони мають високу точність класифікації. На відміну від ранніх систем, які часто могли лише визначати наявність об'єкта, сучасні ЗНМ-архітектури (наприклад, YOLOv8) дозволяють не лише знайти транспортний засіб, а й з високою точністю класифікувати його тип, що експериментально підтверджено у дослідженні [38].

По-третє, ці алгоритми масштабовані та швидкі. Поява систем, таких як YOLO, дозволила виконувати розпізнавання в реальному часі, що є критично

важливим для моніторингу трафіку та систем автономного керування авто. Ефективність цієї моделі у даних задачах доведено у роботах [39] та [40]. Крім того, підходи глибокого навчання легко масштабуються, і точність моделей можна підвищити, збільшивши обсяг тренувальних даних без необхідності змінювати алгоритмічну базу.

Таким чином, перехід від евристичних алгоритмів до глибокого навчання ознаменував зміну парадигми в комп'ютерному зорі, дозволивши створити адаптивні системи, здатні функціонувати в непередбачуваних умовах реального світу з точністю, недосяжною для методів попереднього покоління.

1.2.2 Порівняльний аналіз архітектур двостадійних та одностадійних розпізнавачів

Сучасні методи розпізнавання об'єктів на основі глибокого навчання можна класифікувати на дві основні категорії залежно від їх архітектурного підходу до генерації та класифікації регіонів: двостадійні та одностадійні. Згідно з оглядом архітектур [25], цей поділ базується на наявності або відсутності окремого етапу генерації регіональних пропозицій.

Як детально описано в статті [41], архітектура двостадійних розпізнавачів, до яких належить сімейство R-CNN (R-CNN, Fast R-CNN, Faster R-CNN), характеризується послідовним виконанням двох процесів: генерації набору кандидатів на об'єкти та їх подальшої класифікації й регресії координат. Узагальнена схема цієї архітектури наведена у дослідженні [25] і зображена на рис. 1.5.

Еволюція цього підходу досягла значного прогресу з появою архітектури Faster R-CNN. У роботі [35] зазначається, що на відміну від попередніх версій, що використовували повільні алгоритми вибіркового пошуку, Faster R-CNN інтегрувала спеціалізовану мережу пропозицій регіонів. МПР працює на основі згорткових карт ознак і використовує механізм якорів різного масштабу та співвідношення сторін для генерації потенційних областей, де можуть

знаходиться об'єкти. Це інноваційне рішення дозволило значно підвищити точність локалізації об'єктів .

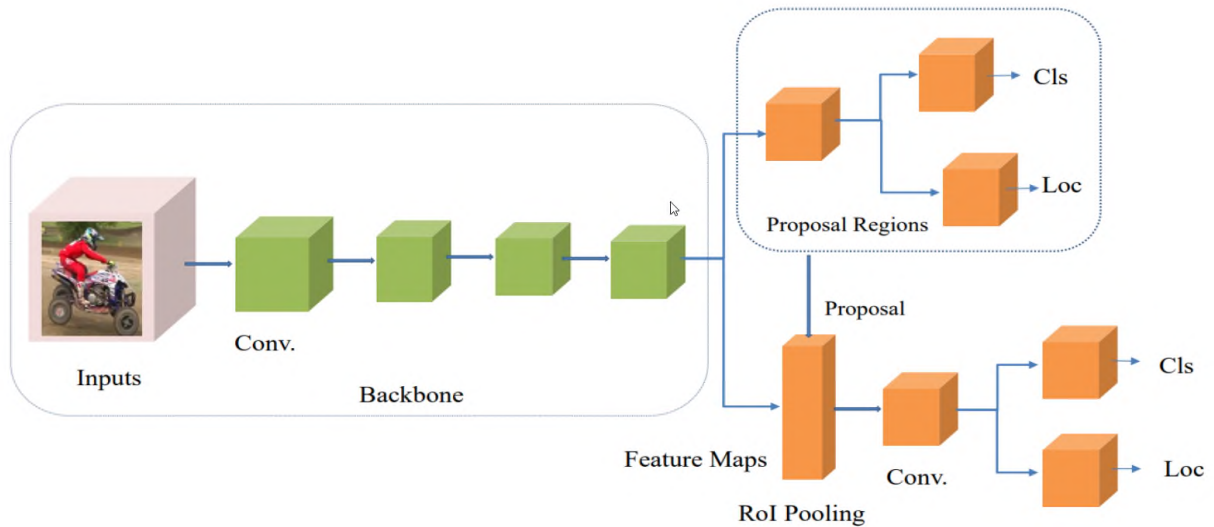


Рисунок 1.5 – Базова архітектура двостадійного розпізнавача [25]

Незважаючи на високу точність, яку демонструють двостадійні моделі завдяки багаторазовому уточненню якорів, їхнім суттєвим недоліком є висока обчислювальна складність. У огляді [25] звертається увага на те, що необхідність виконання кількох проходів для генерації пропозицій та їх класифікації призводить до низької швидкості обробки кадрів, що робить їх менш придатними для систем реального часу, де критичною є затримка обробки .

Одностадійні розпізнавачі, яскравими представниками яких є алгоритми YOLO [34] та SSD (Single Shot MultiBox Detector) [42], пропонують альтернативний підхід, відмовляючись від етапу генерації пропозицій регіонів [25]. В основі цих методів лежить перетворення задачі розпізнавання на задачу регресії, де координати обмежувальних рамок та ймовірності класів передбачаються безпосередньо з вхідного зображення за один прохід нейронної мережі [20]. Архітектура таких систем наведена у дослідженні [25] і показана на рис. 1.6.

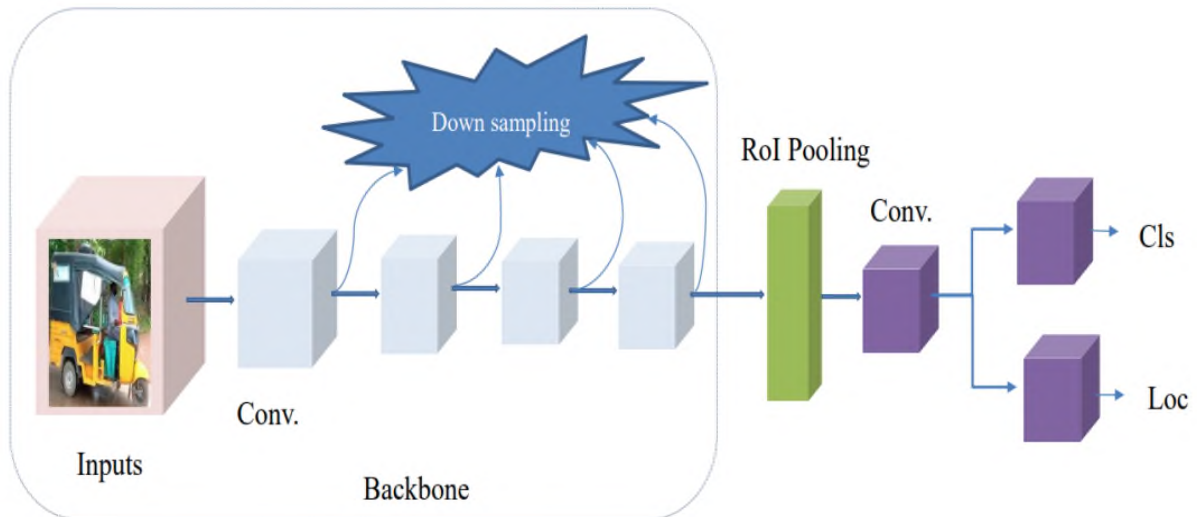


Рисунок 1.6 – Базова архітектура одностадійного розпізнавача [25]

Алгоритм YOLO розділяє вхідне зображення на сітку розміром $S \times S$. Кожна комірка сітки відповідає за розпізнавання об'єкта, якщо центр цього об'єкта потрапляє в дану комірку. Згідно з оглядом [25] мережа одночасно передбачає координати рамок, впевненість у наявності об'єкта та розподіл ймовірностей класів. Такий підхід дозволяє обробляти зображення глобально, враховуючи контекстну інформацію.

Розробник алгоритму SSD зазначає, що їхній підхід використовує згорткову архітектуру прямого поширення, яка генерує набір рамок фіксованого розміру та оцінки для класів об'єктів, використовуючи карти ознак різних масштабів для знаходження об'єктів різного розміру [42].

Головною перевагою одностадійних розпізнавачів є простота оптимізації та висока швидкість генерації припущень щодо об'єктів, що робить їх ідеальними для застосування в мобільних пристроях та системах автономного керування авто.

Ключова відмінність між розглянутими архітектурами полягає у компромісі між швидкістю та точністю.

Згідно з експериментальними даними, одностадійні розпізнавачі значно перевершують двостадійні за швидкістю обробки відеопотоку. Для порівняння, як наведено у дослідженні [25], на наборі даних PASCAL VOC 12 архітектура Faster R-CNN (на базі VGG-16) демонструє швидкість близько 5 кадрів на

секунду, тоді як перша версія YOLO досягає 45, а YOLOv2 – 81 . Це підтверджує, що архітектура YOLO здатна забезпечити роботу в реальному часі, що є критично важливим для моніторингу дорожнього руху. Перевагу у швидкості мережі YOLO описано в огляді [20] та зображено на рис. 1.7.

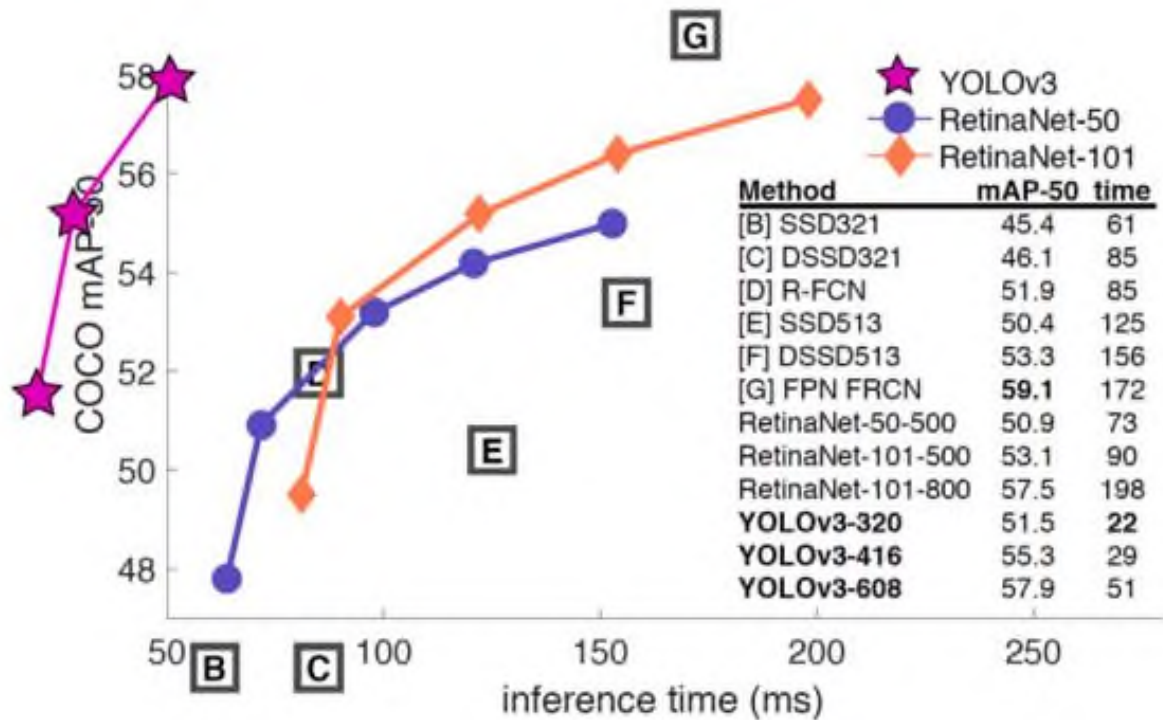


Рисунок 1.7 – Адаптований Дж. Редмонем графік залежності точності розпізнавання від швидкості обробки на наборі COCO [20]

Водночас двостадійні методи, такі як Faster R-CNN, традиційно забезпечують вищу точність локалізації та краще справляються з виявленням дрібних об'єктів, досягаючи вищого показника середньої точності (mAP) на складних наборах даних (67% mAP проти 57,9% у YOLOv1 на PASCAL VOC) . Однак новіші версії одностадійних розпізнавачів (YOLOv3, YOLOv4, YOLOv5) значно скоротили цей розрив у точності, зберігаючи при цьому перевагу у швидкості. У цьому ж дослідженні [25] автор вказує, що YOLOv4 на наборі даних COCO демонструє 43 % AP при 31 FPS, що є оптимальним балансом для сучасних інженерних задач.

Отже, вибір між двостадійними та одностадійними архітектурами залежить від вимог конкретного застосування. Для задач, де пріоритетом є

максимальна точність (наприклад, автономний аналіз аварій), доцільно використовувати Faster R-CNN, тоді як для систем допомоги водію та управління трафіком у реальному часі більш ефективним рішенням є сімейство YOLO.

1.2.3. Архітектурні особливості сімейства YOLO та обґрунтування вибору моделі

Безперервний розвиток сімейства одностадійних розпізнавачів YOLO [34], починаючи з першої версії, представленої у 2015 році призвів до створення високоефективних архітектур, що стали стандартом у задачах розпізнавання об'єктів у реальному часі. Сучасні ітерації, такі як YOLOv5 [43], YOLOv8 [44] та YOLOv11 [45] інтегрували низку передових практик комп'ютерного зору. Як зазначає співавтор архітектури YOLOv7 [46] до таких практик належать методи оптимізації під час тренування, такі як специфічні аугментації даних та багатомасштабне тренування, що суттєво підвищують здатність моделі до узагальнення без збільшення обчислювальних витрат під час висунення припущень.

Сучасні моделі YOLO відрізняються двома фундаментальними архітектурними рішеннями, які роблять їх адаптивними до складних умов.

Першим є перехід до архітектури без якорів. Згідно з оглядом [25] ранні версії YOLO покладалися на попередньо задані якорі для локалізації об'єктів, а ось новітні архітектури, зокрема YOLOv8, використовують підхід без якорів на основі передбачення центру об'єкта. Модель безпосередньо передбачає центр об'єкта та зміщення координат для формування обмежувальної рамки. У контексті моніторингу парковки це критично важливо, так як відмова від фіксованих якорів значно спрощує та покращує розпізнавання об'єктів нестандартних форм та розмірів, наприклад, автомобілів, припаркованих під різними нетиповими кутами відносно камери. У роботі [22] додається, що відмова від якорів робить такі моделі більш ефективними з точки зору обчислень та гнучкими.

Другим є покращене виділення ознак. Для отримання глибоких просторових та семантичних ознак сучасні архітектури використовують оптимізовані блоки. У роботі [22] акцентується на те, що YOLOv8 впроваджує інноваційний модуль C2f, який збагачує градієнтний потік під час зворотного поширення помилки та використовує розділену голову моделі для задач регресії та класифікації. Водночас, як описано в праці [25], в сучасних мережах активно застосовуються модулі просторового пірамідального пулінгу (SPPNet або SPPF), які дозволяють агрегувати локальні ознаки на різних масштабах. Це усуває проблему дублювання градієнтної інформації та мінімізує кількість операцій з плаваючою комою при збереженні високої точності.

Одним із найскладніших завдань візуального моніторингу великих паркувальних майданчиків є розпізнавання транспортних засобів у дальніх рядах, які на зображенні виглядають як об'єкти дуже малого розміру. Сучасні версії YOLO ефективно справляються з цим завданням завдяки використанню мереж пірамідальних ознак [47] та структури PANet [48] у ший моделі. Ця підсистема відповідає за генерацію пірамід ознак, що допомагає мережі знаходити об'єкти різних розмірів та масштабів, поєднуючи висококореневу семантичну інформацію з глибоких шарів із точною просторовою інформацією з ранніх шарів. Завдяки цьому сучасні моделі YOLO демонструють високу точність розпізнавання малих об'єктів та забезпечують низький рівень хибних спрацьовувань. Результат роботи YOLO представлений в огляді та зображений на рис. 1.8.

Аналіз експериментальних даних показує, що для практичної реалізації систем моніторингу існує строгий компроміс між швидкістю та точністю. Оцінюючи ефективність, цей же автор підкреслює, що найбільші моделі сімейства (Large або Extra-Large) вимагають значних обчислювальних потужностей і демонструють при цьому низьку швидкість [20]. Натомість полегшені версії Nano (n) та Small (s) забезпечують високу швидкість обробки кадрів. Дослідження показують, що при переході від версії Large до версій Nano або Small втрата загальної точності становить лише близько 0,8–1,3 %, тоді як швидкість обробки зростає в кілька разів, досягаючи 47 кадрів на секунду [20].

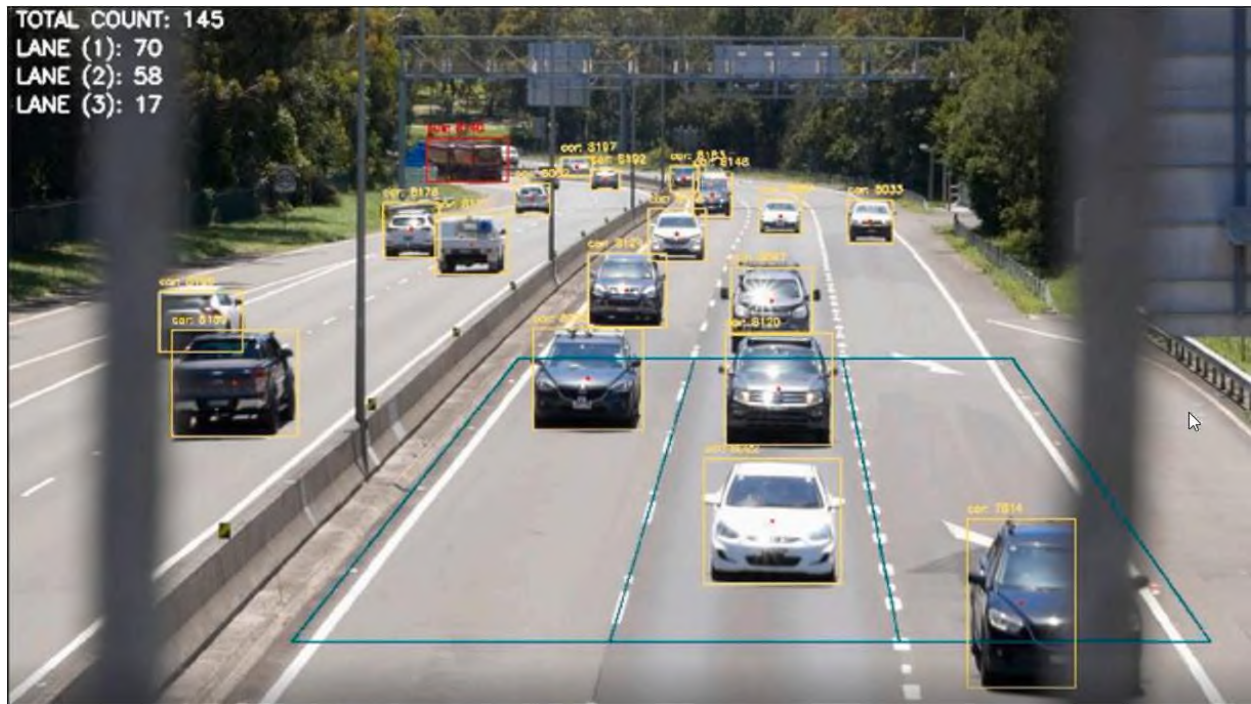


Рисунок 1.8 – Центроїди і підписи авто на різних відстанях [20]

Важливою практичною деталлю також є те, що сучасні важкі моделі для забезпечення високої швидкості обробки в режимі реального часу вимагають наявності спеціалізованих прискорювачів (GPU або TPU). Це унеможливорює розгортання цих моделей на звичайних офісних комп'ютерах, які найчастіше використовуються на постах охорони (де в цілому і необхідне впровадження таких систем), адже вони не оснащені потужними відеокартами. Саме тому ключовою перевагою використання полегшених архітектур YOLO Nano або Small є їхня здатність забезпечувати достатній рівень кадрової частоти під час виконання обчислень у моделі безпосередньо на центральному процесорі. Це є критичною вимогою для розробки бюджетної системи моніторингу, оскільки усуває необхідність закупівлі дороговартісного серверного обладнання.

1.3 Аналіз методів просторової фільтрації та просторового аналізу зображень

Сучасні моделі глибокого навчання демонструють визначну швидкість і точність у вирішенні задачі локалізації об'єктів на зображенні. Як зазначають у

роботі [49], результатами роботи таких неймереж є формування обмежувальних рамок навколо знайдених об'єктів, визначення їхнього класу та розрахунок рівня впевненості моделі. Проте ці алгоритми за своєю природою вирішують виключно базову задачу розпізнавання у двовимірній площині кадру і не надають жодної інформації про семантичне розташування об'єкта в реальному просторі. Для систем розумного паркування простої наявності автомобіля в кадрі недостатньо. Системі необхідно здійснювати точне просторове співставлення виявленого транспортного засобу із конкретним цільовим паркомісцем. У дослідженні [50] підкреслюється, що такий підхід є обов'язковим для ігнорування автомобілів, які просто проїжджають повз, або тих транспортних засобів, що припарковані з порушеннями ліній розмітки чи за межами дозволених зон. Відповідно, після етапу роботи неймережі система потребує застосування алгоритмів просторової фільтрації на основі виділених зон інтересу.

Найпростішим та найменш обчислювально затратним геометричним підходом до просторового співставлення є метод центроїда. Кожне паркомісце моделюється як багатокутник, координати вершин якого заздалегідь прописуються в конфігурації системи. Для кожної обмежувальної рамки $B = [x_1, y_1, x_2, y_2]$, яку згенерувала модель, обчислюється її центральна точка (центроїд) $P(x, y)$ за простими формулами $x = \frac{x_1 + x_2}{2}$ та $y = \frac{y_1 + y_2}{2}$. Відповідно до методології, запропонованої у роботі [49], далі алгоритм (наприклад, з використанням функції `cv2.pointPolygonTest`) перевіряє факт входження цього центроїда всередину заданого багатокутника паркомісця. Схожий підхід розглядається у статті [51], де аналогічна логіка застосовується і при накладанні попиксельних масок постобробки. Якщо центральна координата рамки автомобіля збігається з корисним (чорним) пікселем маски зони інтересу, об'єкт зараховується як припаркований.

Незважаючи на легкість реалізації, цей метод має вагомні недоліки. Він є чутливим до неточностей обмежувальних рамок. Якщо неймережа виділить рамку, що охоплює автомобіль не повністю, або якщо авто припарковане під

кутом, центроїд може зміститися за межі багатокутника, що призведе до хибного висновку про вільне паркомісце.

Більш надійним та стійким до похибок розпізнавання є підхід на основі перекриття площ та метрики IoU. Замість використання однієї центральної точки цей метод аналізує площу перекриття між згенерованою рамкою автомобіля (B) та багатокутником паркувальної зони (R). Метрика IoU розраховується як відношення площі перетину цих двох фігур до площі їхнього об'єднання
$$IoU(B, R) = \frac{|B \cap R|}{|B \cup R|}$$
. Використання метрики IoU дозволяє встановити жорстке порогове значення (наприклад, $T > 0,5$). Якщо площа перекриття перевищує цей поріг, паркомісце вважається зайнятим [49]. Такий поріг є ефективним механізмом фільтрації хибних спрацювань. У випадку, коли автомобіль проїжджає по краю паркувального слота або припаркований із суттєвим порушенням (займаючи два місця одночасно), його рамка не створить достатнього перекриття з цільовим полігоном (хоча другий випадок потребує більш детального аналізу). Автори роботи [51] доводять, що комбінація IoU із методом постобробки масок (де маска накладається не на вхідне зображення, а застосовується до координат результатів розпізнавання) дозволяє зберегти контекст зображення для нейромережі, при цьому чітко відфільтровуючи об'єкти на фоні та поза межами зон інтересу.

На практиці просторовий аналіз часто стикається з масштабнішими проблемами: візуальними спотвореннями від об'єктів камер, перспективою та необхідністю ручного розмічання зон інтересу. У роботі [50] звертають увагу на те, що ручна розмітка кожного паркомісця є трудомістким процесом, що обмежує можливість легкого масштабування систем на нові локації. Для вирішення проблеми автоматизації розмітки інноваційним рішенням є генерація теплових карт центроїдів. Як запропоновано в роботі [52], генерація теплових карт ґрунтується на гіпотезі, що паркомісця – це специфічні зони, де транспортні засоби залишаються нерухомими протягом тривалого періоду часу. Використовуючи моделі сегментації екземплярів, система протягом певного часу накопичує маски знайдених автомобілів. Накладаючи ці результати один на одного, алгоритм формує теплову карту, де гарячі зони відповідають реальним

місцям паркування. Автор статті [50] пояснює, що після їх побудови система здатна автоматично обчислювати обернені прямокутники для знайдених гарячих зон, повністю позбавляючи оператора необхідності ручного введення координат полігонів.

Іншою вагомою проблемою є застосування ширококутних камер типу риб'яче око. У роботі [53] зазначають, що такі камери, з одного боку, охоплюють набагато більшу площу (до 195 градусів), що усуває сліпі зони, але з іншого – вносять сильне радіальне спотворення. Традиційні алгоритми компенсації, такі як обернене перспективне перетворення, генерують значні геометричні похибки, які спотворюють розміри та локалізацію об'єктів на великій відстані, штучно обмежуючи радіус дії систем паркування.

Ефективним вирішенням проблеми перспективи є проєкція вигляду зверху з використанням архітектури крос-видових трансформерів. У згаданому дослідженні пропонується підхід, у якому ознаки, витягнуті з кількох таких камер, проходять через механізм уваги і проєктуються у єдину репрезентативну сітку у просторі [53]. Автори також зауважують, що при цьому радіальне спотворення оптики компенсується специфічними позиційними вбудовуваннями, розрахованими на основі поліноміальної моделі [53]. Отримані практичні результати підтверджують, що сформована таким чином карта дозволяє знаходити паркомісця та орієнтацію автомобілів як єдине ціле, без артефактів склеювання чи спотворення високих об'єктів, досягаючи похибки позиціонування менше 23 см у просторі 25 на 25 метрів [53].

Попри очевидні переваги інноваційних методів просторового аналізу, таких як автоматична генерація теплових карт за допомогою моделей сегментації або побудова проєкцій на основі крос-видових трансформерів, їхнє практичне застосування має суттєвий недолік – високі вимоги до обчислювальних ресурсів. Запуск таких архітектур у реальному часі на звичайних центральних процесорах є практично неможливим. Враховуючи, що головною метою даної роботи є вдосконалення існуючих методів визначення зайнятості паркувальних місць для роботи в умовах обмежених ресурсів стандартного ПК, найоптимальнішим компромісом між достатньою точністю та низькою обчислювальною складністю

залишається використання базового методу просторової фільтрації – комбінації накладання попередньо визначених зон інтересу та розрахунку метрики перекриття площ. Вдосконалення саме цього підходу дозволить забезпечити надійну роботу системи на апаратному забезпеченні низького рівня.

Для створення високоефективної системи комп'ютерного зору у сфері розумного паркування недостатньо покладатися виключно на можливості неймереж. Синергічна комбінація точного сучасного розпізнавача (наприклад, YOLO) із надійною логікою просторової фільтрації (застосування масок зон інтересу у поєднанні з метрикою перекриття) є ключем до створення безвідмовної архітектури. А впровадження інноваційних методів просторового аналізу, таких як автоматична генерація зон інтересу на основі теплових карт та проєкцій для компенсації спотворень лінз, дозволяє вивести систему на новий рівень автоматизації та масштабованості при доступності додаткових обчислювальних ресурсів.

1.4 Висновки і постановка задачі дослідження

У першому розділі проведено комплексний аналіз сучасного стану технологій та методів візуального моніторингу паркувального простору.

Аналіз ринку апаратно-програмних комплексів показав, що існуючі комерційні рішення (апаратно-орієнтовані системи закритого типу, хмарні платформи та рішення з граничними обчисленнями) мають суттєві обмеження для масового впровадження. Вони характеризуються високою вартістю інфраструктури, явищем залежності від постачальника та стабільності інтернет-з'єднання. Це обґрунтовує необхідність створення гнучкої локальної кіберфізичної системи на базі стандартних IP-камер та універсальних обчислювальних пристроїв.

Огляд еволюції методів комп'ютерного зору підтвердив неефективність традиційних евристичних алгоритмів (ГНГ, каскади Хаара) у складних вуличних умовах. Порівняльний аналіз архітектур глибокого навчання (двостадійні та одностадійні розпізнавачі) довів, що для задач реального часу

найоптимальнішим вибором є сімейство моделей YOLO. Зокрема, використання легких версій (YOLOv8 Nano/Small) з архітектурою без якорів дозволяє досягти балансу між високою точністю та достатньою швидкістю.

Аналіз методів просторового співставлення виявив, що базове розпізнавання об'єктів є недостатнім для визначення статусу паркомісця. Для усунення хибних спрацювань необхідно застосовувати алгоритми просторової фільтрації. Комбінація методів накладання зон інтересу та розрахунку площі перекриття визнана найбільш надійним підходом для прив'язки знайденого авто до конкретного місця.

Отже, проблема нашого дослідження зумовлена об'єктивними вимогами щодо необхідності зниження апаратної вартості систем моніторингу паркувального простору при одночасному збереженні високої точності розпізнавання та стійкості до перешкод у складних вуличних умовах (перекриття об'єктів, зміни освітлення, геометричні спотворення). Зменшити ступінь цих суперечностей через розроблення відповідних програмних засобів має намір наше дослідження. Обґрунтуванням необхідності такого впливу є пряме підвищення ефективності та економічної доцільності розгортання інтелектуальних транспортних систем.

З огляду на вищезазначене, метою дипломної роботи є вдосконалення існуючих методів для визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторового аналізу відеозображень для забезпечення їх роботи в умовах обмежених обчислювальних ресурсів.

Для досягнення поставленої мети необхідно вирішити такі задачі дослідження:

- проаналізувати сучасний стан, існуючі апаратно-програмні комплекси та методи комп'ютерного зору у сфері візуального моніторингу паркувального простору;
- розробити математичні моделі просторового відбору об'єктів та часової стабілізації станів для компенсації ефекту візуального паралаксу;

- удосконалити метод просторово-часового аналізу відеоданих шляхом впровадження адаптивної метрики просторового перекриття та алгоритмів фільтрації на базі скінченного автомата з подвійним порогом;
- спроектувати та розробити програмне забезпечення кіберфізичної системи з урахуванням архітектурної вимоги щодо виконання нейромережевого прогнозування на центральному процесорі;
- провести експериментальні дослідження для оцінки точності розпізнавання та ресурсної ефективності оптимізованої системи в реальних умовах експлуатації.

2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ВИЗНАЧЕННЯ ЗАЙНЯТОСТІ ПАРКОМІСЦЬ

2.1 Обґрунтування вибору методу та розробка загальної концепції системи

На основі аналізу, проведеного у першому розділі роботи, встановлено, що найефективнішим інструментом для вирішення задачі візуального моніторингу в режимі реального часу є застосування одностадійних розпізнавачів об'єктів сімейства YOLO (зокрема, полегшених версій архітектур YOLOv8, v11) [45, 51]. Вибір даного інструменту обґрунтовується його здатністю ефективно виконувати обчислення на центральних процесорах без критичної втрати точності розпізнавання. Для розробки кіберфізичної системи формалізовано математичний апарат, який дозволяє об'єктивно оцінювати якість роботи обраної моделі, та описано концепцію взаємодії нейромережі з алгоритмами просторового аналізу.

Оскільки розроблювана кіберфізична система має працювати в умовах реального часу з мінімальною кількістю хибних спрацювань, для об'єктивної оцінки якості роботи нейромережі було обрано метрики Precision, Recall та mAP. Базою для їх розрахунку є матриця невідповідностей, яка оперує наступними базовими поняттями:

- *TP* (True Positive – істинно позитивний результат) – кількість коректно локалізованих та класифікованих об'єктів;
- *FP* (False Positive – хибно позитивний результат) – кількість об'єктів, які система помилково розпізнала як потрібні об'єкти;
- *FN* (False Negative – хибно негативний результат) – кількість реальних потрібних об'єктів, які система не змогла виявити;
- *TN* (True Negative – істинно негативний результат) – кількість випадків, де модель не виявила об'єкт.

У задачах розпізнавання *TN* використовується дуже рідко, тому що на зображенні завжди присутня майже нескінчена кількість регіонів, де немає об'єктів. Проте в контексті задачі кваліфікаційної роботи *TN* відповідає коректно розпізаному вільному паркувальному слоту.

Рішення про те, чи є розпізнавання істинним (TP), приймається на основі метрики просторового перекриття IoU між передбаченою обмежувальною рамкою B_{pred} та еталонною рамкою B_{gt} , розміченою експертом або алгоритмом на основі теплової карти. IoU розраховується за формулою:

$$IoU = \frac{Area(B_{pred} \cap B_{gt})}{Area(B_{pred} \cup B_{gt})} \quad (2.1)$$

де $Area(B_{pred} \cap B_{gt})$ – площа перетину передбаченої та еталонної рамок;

$Area(B_{pred} \cup B_{gt})$ – площа об'єднання передбаченої та еталонної рамок.

Якщо розраховане значення IoU перевищує заданий поріг T_{iou} , локалізація вважається успішною (TP). На основі цих базових значень розраховуються головні метрики якості моделі.

Точність (Precision) відображає частку істинних об'єктів серед об'єктів, які система класифікувала як потрібні. Ця метрика показує здатність алгоритму не генерувати хибні спрацювання. Точність визначається за виразом:

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

Повнота (Recall) відображає частку знайдених об'єктів від їх загальної реальної кількості на сцені. Метрика характеризує здатність моделі знаходити всі цільові об'єкти. Формула (2.3) описує співвідношення для знаходження повноти:

$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

Для повної оцінки якості системи розраховується точність класифікації системи (Accuracy). Точність класифікації обчислюється за формулою:

$$Accuracy = \frac{TN + TP}{TP + FP + TN + FN}. \quad (2.4)$$

Для глибокого аналізу чутливості та специфічності алгоритмів прийняття рішень застосовуються метрики частки істинно позитивних (TPR – True Positive Rate) та хибно позитивних (FPR – False Positive Rate) класифікацій. Значення TPR аналогічне Recall. TPR та FPR визначаються за співвідношеннями:

$$TPR = \frac{TP}{TP + FN}. \quad (2.5)$$

$$FPR = \frac{FP}{FP + TN}. \quad (2.6)$$

Ці показники є базовими для побудови ROC-кривої, яка ілюструє якість двійкової класифікації при різних порогових значеннях. Площа під цією кривою (AUC) оцінюється за наближеною формулою:

$$AUC = \frac{1}{2} - \frac{FPR}{2} + \frac{TPR}{2}. \quad (2.7)$$

Для комплексної оцінки балансу між точністю та повнотою обчислюється гармонійне середнє – F_1 -оцінка за формулою:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad (2.8)$$

Головною інтегральною метрикою для розпізнавачів об'єктів є середня точність (AP). Вона обчислюється як площа під кривою Precision-Recall (PR-кривою) для кожного класу об'єктів. У задачі з виявленням одного класу (транспортний засіб) розрахунок відбувається за інтегралом:

$$AP = \int_0^1 P(Re)dr, \quad (2.9)$$

де $P(Re)$ – значення точності як функція від повноти Re .

Цей інтеграл розраховується наближено числовими методами, такими, як метод трапецій, підставляючи значення повноти від 0 до 1 з кроком 0,1. Таким чином, інтеграл замінюється обчисленням площі з інтерполяцією відповідно до виразу:

$$AP = \sum_n (Re_{n+1} - Re_n) \max_{Re: \hat{Re} \geq Re_{n+1}} P(\hat{Re}), \quad (2.10)$$

де Re та P – значення повноти та точності відповідно на n -му кроці обчислень.

Задача розпізнавання зайнятості паркомісця має лише один клас – транспортний засіб. У випадку подальшого масштабування системи та розширення її функціоналу на розпізнавання декількох класів об'єктів (наприклад, відокремлення легкових автомобілів, вантажівок та мотоциклів), використовується усереднена середня точність (mAP – mean Average Precision), що є середнім арифметичним значень AP для кожного з n класів і розраховується за виразом:

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i. \quad (2.11)$$

У задачі з одним класом $mAP = AP$.

З огляду на обрані критерії оцінки, концепція запропонованої кіберфізичної системи формалізується наступним чином. Загальна задача визначення зайнятості паркувального майданчика зводиться до відображення множини пікселів вхідного зображення на вектор логічних станів паркомісць. Нехай I — двовимірна матриця пікселів поточного відеокадру. Паркувальний майданчик описується як множина статичних просторових зон інтересу (полігонів) S , де кожне паркомісце $s_i \in S$, $i =$

$1, 2, \dots, N$ (N – загальна кількість паркомісць). Процес обробки поділяється на дві послідовні функції. Перша функція – це функція нейромережевого розпізнавання f_{det} , яка приймає на вхід зображення I та повертає множину виявлених обмежувальних рамок транспортних засобів B . Ця функція описується виразом:

$$f_{det}(I) \rightarrow B = \{b_1, b_2, \dots, b_M\}, \quad (2.12)$$

де M – кількість знайдених автомобілів на кадрі.

Кожна рамка b_j описується координатами своїх кутів у двовимірному просторі кадру.

Друга функція – це функція просторового аналізу f_{space} , яка перевіряє геометричне співставлення кожної знайденої рамки b_j до кожного паркомісця s_i . У формулі (2.13) описано множину значень функції.

$$f_{space}(b_j, s_i) \rightarrow \{0, 1\}. \quad (2.13)$$

Ця функція повертає 1, якщо автомобіль геометрично знаходиться в межах паркувального слота, і 0 – якщо ні.

Таким чином, фінальним результатом роботи системи для кожного поточного моменту часу $t \in \mathbb{N}$ є формування вектора станів паркувального майданчика $Y_{occ}(t)$. Вектор описується виразом:

$$Y_{occ}(t) = [y_{occ,1}, y_{occ,2}, \dots, y_{occ,N}], \quad (2.14)$$

де $y_{occ,i} \in \{0,1\}$ – логічний стан i -го паркомісця (0 – вільно, 1 – зайнято);

$t \in \mathbb{N}$ - індекс поточного такту обчислень (відеокадру).

Запропонована концептуальна математична модель розділяє задачу на підзадачу локалізації об'єктів та підзадачу просторової фільтрації.

2.2 Математична модель сцени паркування, геометричних перетворень та розпізнавання об'єктів

2.2.1 Математична модель сцени

У процесі візуального моніторингу цифрова камера виконує проєкцію реальної тривимірної сцени (паркувального майданчика) на двовимірну площину матриці сенсора. Оскільки оптична вісь камери зазвичай розташована під певним кутом до площини дороги, виникає перспективне спотворення. Паралельні лінії перетинаються на зображенні, а об'єкти на задньому плані здаються меншими, ніж об'єкти на передньому. Формалізуємо геометрію перспективної проєкції при формуванні зон інтересу.

Нехай $I \in \mathbb{R}^{W \times H \times C}$ – вхідний кадр відеопотоку, де \mathbb{R} – множина дійсних чисел, W та H – ширина та висота зображення в пікселях, а C – кількість кольорових каналів (для простору RGB $C=3$).

Введемо глобальну декартову систему координат зображення. Початок координат $(0, 0)$ розташований у лівому верхньому куті кадру. Вісь абсцис X спрямована праворуч, а вісь ординат Y – вниз. Тоді будь-який піксель на зображенні описується вектором-стовпцем своїх координат $p = [x, y]^T$, де $0 \leq x < W$ та $0 \leq y < H$.

У зв'язку з описаним вище перспективним спотворенням реальне прямокутне паркувальне місце на двовимірному зображенні набуває форми довільного опуклого багатокутника. Тому i -те паркомісце моделюється як впорядкована множина вершин (вузлів) у системі координат зображення:

$$P_i = \{V_{i,1}, V_{i,2}, \dots, V_{i,K}\}, \quad (2.15)$$

де K – кількість вершин полігона (найчастіше для паркомісць використовується чотирикутник, $K = 4$);

$V_{i,k} = (x_k, y_k)$ – координати k -ї вершини i -го паркомісця.

Оскільки багатокутник є замкненою геометричною фігурою, відрізки, що з'єднують сусідні вершини $V_{i,k}$ та $V_{i,k+1}$ (а також останню з першою), утворюють його межі. Будь-яка точка $p(x, y)$, що лежить усередині простору, обмеженого цими границями, вважається такою, що належить паркомісцю P_i .

Нейронна мережа YOLO, виконуючи задачу локалізації, описує видимі межі транспортного засобу прямокутною обмежувальною рамкою (Bounding box). Результат розпізнавання j -го автомобіля описується кортежем з чотирьох координат:

$$b_j = (x_{min}, y_{min}, x_{max}, y_{max}), \quad (2.16)$$

де (x_{min}, y_{min}) – координати верхнього лівого кута рамки;

(x_{max}, y_{max}) – координати нижнього правого кута.

Для розв'язання задачі просторового співставлення, а саме визначення входження об'єкта в багатокутник, представлення рамки через крайні точки є незручним. Більш релевантними є параметричне представлення рамки через координати її геометричного центру $C(x_c, y_c)$ та лінійні розміри – ширину w та висоту h . Обрахунок цих координат і величин відбувається за виразами:

$$x_c = \frac{x_{min} + x_{max}}{2}, y_c = \frac{y_{min} + y_{max}}{2}. \quad (2.17)$$

$$w = x_{max} - x_{min}, h = y_{max} - y_{min}. \quad (2.18)$$

На рис. 2.1 зображено геометричну формалізацію сцени. Паркомісце позначається багатокутником P_i з вершинами $V_{i,1} - V_{i,4}$, а розпізнане авто прямокутником b_j з позначеним центром $C(x_c, y_c)$.

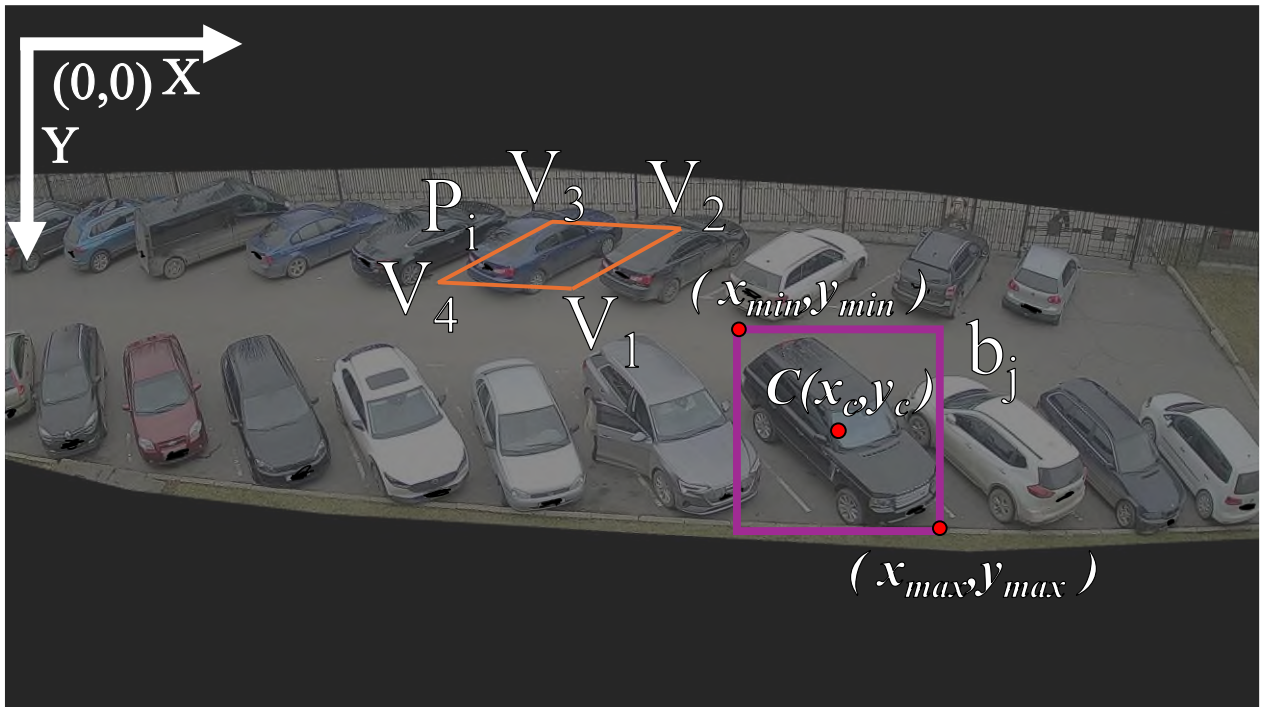


Рисунок 2.1 – Геометрична формалізація сцени
(полігон P_i та обмежувальна рамка b_j з точкою C)

При роботі з зображеннями, які є плоскими проєкціями складних сцен виникає ефект паралаксу. Якщо камера встановлена під гострим кутом, високі об'єкти (наприклад, дах позашляховика) візуально знаходяться на сусідніх паркувальних місцях на зображенні, хоча колеса автомобіля знаходяться в межах свого слота.

Перед тим як аналізувати двовимірні координати на зображенні, формалізуємо процес проєкції реального тривимірного простору на площину кадру. Для цього використаємо просторову модель камери-обскури. Проєкція довільної 3D-точки $M(X, Y, Z)$ світової системи координат у 2D-точку $m(u, v)$ на площині зображення описується наступним матричним рівнянням:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [R \quad t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.19)$$

де s – масштабний множник;

K – матриця внутрішніх параметрів камери (фокусна відстань та координати оптичного центру);

R та t – матриця обертання та вектор трансляції (зовнішні параметри, що визначають положення камери над паркувальним майданчиком).

З цього рівняння випливає природа ефекту візуального паралаксу. Для точок об'єкта, що знаходяться безпосередньо на рівні асфальту (колеса автомобіля, де апліката $Z \approx 0$), їх проєкція (u, v) строго відповідає плоскому полігону паркомісця. Однак для точок даху транспортного засобу ($Z = h > 0$), їх проєкція на матрицю камери суттєво зміщується вздовж променя зору.

Ця просторова закономірність доводить хибність використання геометричного центру 2D-рамки об'єкта $C(x_c, y_c)$ (який часто зміщений до даху) як ідентифікатора його реального положення на площині.

Для компенсації цієї похибки без використання складних 3D-перетворень введемо поняття опорної точки або центру нижньої грані обмежувальної рамки. Координати цієї точки розраховуються за виразом:

$$V_{base} = (x_c, y_{max}) = \left(\frac{x_{min} + x_{max}}{2}, y_{max} \right). \quad (2.20)$$

Точка V_{base} є апроксимацією місця контакту транспортного засобу з площиною дороги. Використання цієї точки для подальшого просторового аналізу дозволяє нівелювати вплив висоти автомобіля та перспективних спотворень на точність визначення зайнятості конкретного паркомісця.

На рис. 2.2 зображено приклад такої ситуації. Чорний автомобіль з високим дахом має велику обмежувальну рамку, через що геометричний центр C перебуває на межі двох паркувальних місць. Це призводить до помилкового визначення зайнятого слота. На відміну від точки C , точка V_{base} коректно знаходиться на потрібному паркомісці.

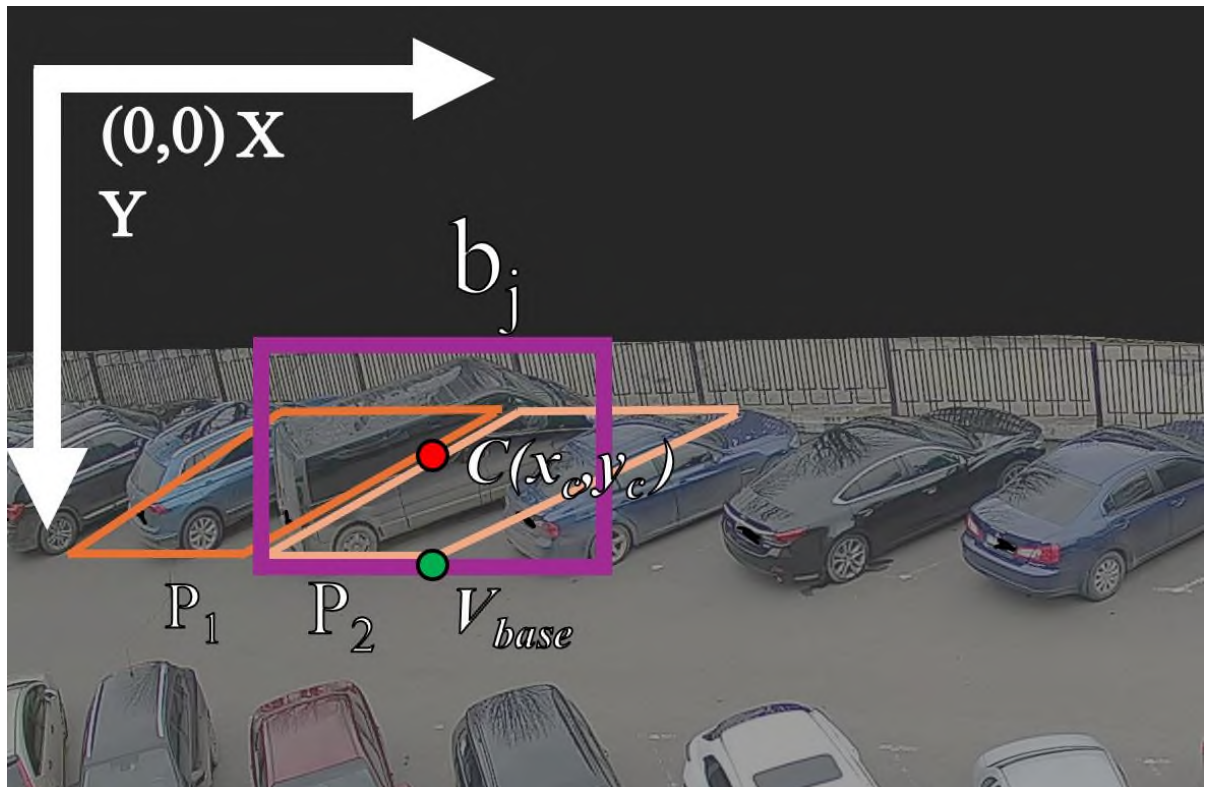


Рисунок 2.2 – Компенсація ефекту паралаксу

2.2.2 Математична модель фільтрації множини розпізнаних об'єктів

Нейронна мережа YOLO формує передбачення у вигляді густої сітки, що призводить до генерації великої кількості обмежувальних рамок для одного і того самого фізичного об'єкта. Для коректної роботи системи аналізу математично формалізуємо процес зменшення цієї надлишкової множини до мінімально необхідної, де кожному реальному об'єкту відповідатиме рівно одна рамка. Цей процес реалізується за допомогою алгоритму пригнічення немаксимумів.

Нехай на виході нейромережі сформовано початкову множину розпізнаних об'єктів D . Кожен елемент цієї множини $d_j \in D$ являє собою кортеж:

$$d_j = (b_j, c_j, l_j), \quad (2.21)$$

де b_j – геометричні координати рамки;

а $c_j \in [0,1]$ – оцінка впевненості моделі у тому, що в даній рамці дійсно знаходиться певний клас об'єкту;

а $l_j \in L$ – ідентифікатор цього класу (L – повна множина класів, які розпізнає модель).

Для усунення хибних спрацювань на об'єкти, що не стосуються паркування (наприклад, пішоходи чи тварини), у математичній моделі виділено підмножину цільових класів $L_{target} \subset L$, яка включає лише транспортні засоби (наприклад $L_{target} = \{\text{"машина"}, \text{"мотоцикл"}, \text{"автобус"}, \text{"вантажівка"}\}$).

Процес фільтрації складається з двох етапів. На першому етапі застосовується жорстке порогове відсікання за впевненістю та фільтрація за класом об'єкта. Формується проміжна множина D_{conf} :

$$D_{conf} = \{d_j \in D \mid c_j \geq T_{conf} \wedge l_j \in L_{target}\}, \quad (2.22)$$

де T_{conf} – мінімально допустимий поріг впевненості.

На другому етапі застосовується ітеративний алгоритм пригнічення не максимумів до множини D_{conf} . Алгоритм ініціалізує порожню множину відфільтрованих результатів $D_{final} = \emptyset$. На кожній ітерації з множини D_{conf} вилучається елемент $d_{max} = (b_{max}, c_{max}, l_{max})$ з найбільшим значенням c_j і додається до D_{final} . Вирази (2.23 – 2.25) описують визначення цього елемента і перетворення множин.

$$d_{max} = \arg \max_{d_j \in D_{conf}} c_j. \quad (2.23)$$

$$D_{final} = D_{final} \cup \{d_{max}\}. \quad (2.24)$$

$$D_{conf} = D_{conf} \setminus \{d_{max}\}. \quad (2.25)$$

Далі множина (2.26) D_{conf} оновлюється шляхом видалення всіх рамок, які мають занадто велике просторове перекриття з щойно обраною рамкою b_{max} . Для цього використовується метрика IoU :

$$D_{conf} = \{d_j \in D_{conf} \mid IoU(b_j, b_{max}) < T_{nms}\}, \quad (2.26)$$

де T_{nms} – поріг пригнічення немаксимумів.

Цей ітеративний процес повторюється доти, доки множина D_{conf} не стане порожньою. Результатом роботи алгоритму є множина D_{final} , елементи якої передаються у модуль просторової селекції.

2.3 Математичне моделювання просторового відбору об'єктів на основі полігональних масок

Ключовим етапом запропонованого просторового аналізу є перевірка входження обчисленої опорної точки $V_{base}(x_{base}, y_{base})$ у межі багатокутника паркомісця [58]. Для цього застосовано алгоритм трасування променя на основі правила парності перетинів.

2.3.1 Перевірка належності точки багатокутнику

Універсальний алгоритм трасування променя вирішує задачу входження.

Суть алгоритму полягає у проведенні горизонтального променя з досліджуваної точки V_{base} в додатному напрямку осі X ($x \rightarrow +\infty$) та підрахунку кількості перетинів цього променя з ребрами багатокутника P_i . Теоретичним обґрунтуванням цього методу є топологічна теорема Жордана про замкнену криву, яка ділить площину на внутрішню та зовнішню області. Наслідком цієї теореми є те, що промінь, випущений з точки всередині багатокутника у нескінченність (яка гарантовано належить зовнішній області), повинен перетнути межу фігури непарну кількість разів. Відповідно, парна кількість перетинів (або їх відсутність) свідчить про те, що початкова точка знаходиться зовні.

Багатокутник P_i (2.15) задано впорядкованою множиною з K вершин. Ребро багатокутника E_k утворюється відрізком між вершинами $V_{i,k}(x_k, y_k)$ та $V_{i,k+1}(x_{k+1}, y_{k+1})$.

Для замикання контуру приймається, що $V_{i,K+1} = V_{i,1}$. Умова перетину горизонтального променя, випущеного з точки (x_{base}, y_{base}) , з ребром E_k виконується тоді, коли ордината точки (2.27) знаходиться строго між ординатами вершин ребра.

$$y_m \in [\min(y_k, y_{k+1}), \max(y_k, y_{k+1})]. \quad (2.27)$$

Якщо ця умова виконується, необхідно знайти абсцису точки перетину променя з ребром x_{int} за рівнянням прямої (2.28), що проходить через дві задані точки.

$$x_{int} = x_k + \frac{(y_{base} - y_k)(x_{k+1} - x_k)}{y_{k+1} - y_k}. \quad (2.28)$$

Якщо розрахована координата $x_{int} > x_m$, промінь перетинає дане ребро, і лічильник перетинів C збільшується. Після перевірки всіх K ребер багатокутника приймається логічне рішення щодо просторової належності об'єкта (функція f_{pip}). Функція описується системою:

$$f_{pip}(V_{base}, P_i) = \begin{cases} 1, & \text{якщо } C \pmod{2} \neq 0 \\ 0, & \text{якщо } C \pmod{2} = 0 \end{cases} \quad (2.29)$$

При реалізації враховується ситуація, де ребро є горизонтальним ($y_{k+1} = y_k$), тому що при цьому знаменник вироджується до нуля.

Однією із особливостей цього методу є те, що він працює для будь-яких багатокутників, не обов'язково опуклих. Проте, враховуючи специфіку предметної області, можна застосувати алгоритмічну оптимізацію. Оскільки реальне паркувальне місце є найчастіше паралелограмом, його перспективна проєкція на площину кадру камери гарантовано утворює опуклий багатокутник.

На рис. 2.3 зображено промені з різних точок, зовні та всередині п'ятикутника.

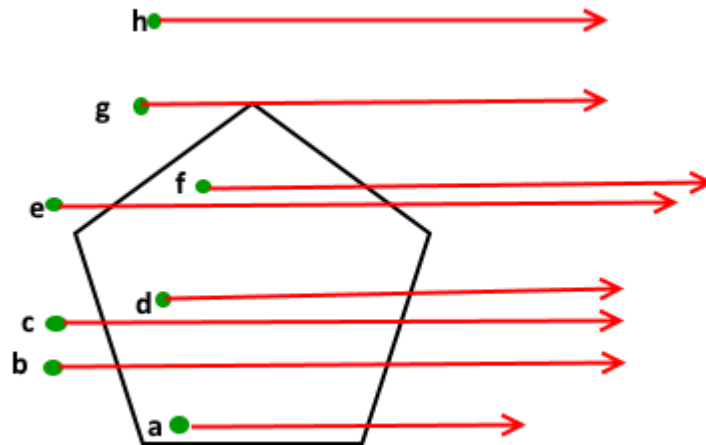


Рисунок 2.3 – Алгоритм трасування променів

Для строго опуклих багатокутників найефективнішим методом перевірки входження точки є метод півплощин (або метод однакових знаків псевдоскалярних добутків). Його суть полягає в тому, що точка знаходиться всередині опуклого багатокутника тоді і тільки тоді, коли вона лежить по одному і тому самому боці від усіх його спрямованих ребер (при послідовному обході вершин в одному напрямку). На рис. 2.4 зображено як будуть виглядати ці вектори при точці всередині багатокутника (a) і зовні (b).

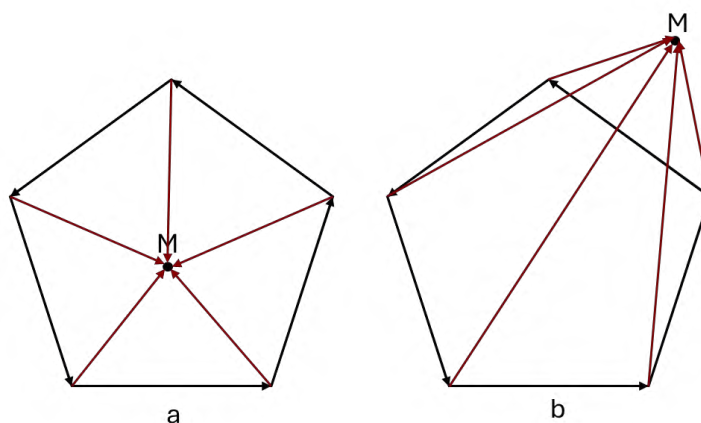


Рисунок 2.4 – Метод з використанням косоного добутку

Цей спосіб використовує ті ж K вершин багатокутника, обхід яких здійснюється за годинниковою (або проти) стрілкою. Для замикання контуру вважаємо, що $V_{i,k+1} = V_{i,k}$.

Для кожного k -го ребра багатокутника формується спрямований вектор грані $E_k = V_{i,k+1} - V_{i,k}$ та вектор від початкової вершини грані до досліджуваної точки $W_k = V_{base} - V_{i,k}$.

Для визначення взаємного розташування вектора грані та точки обчислюється їх двовимірний псевдоскалярний (косий) добуток, який обчислюється за формулою:

$$\Delta_k = E_k \times W_k = (x_{k+1} - x_k)(y_{base} - y_k) - (x_{base} - x_k)(y_{k+1} - y_k). \quad (2.30)$$

Значення Δ_k геометрично відображає орієнтовану площу паралелограма, побудованого на цих векторах.

Якщо $\Delta_k > 0$, точка V_{base} лежить ліворуч від спрямованого ребра.

Якщо $\Delta_k < 0$, точка лежить праворуч від ребра.

Якщо $\Delta_k = 0$, точка лежить безпосередньо на лінії, що містить ребро.

Функція просторової належності $f_{rip}(V_{base}, P_i)$ формується на основі перевірки збігу знаків для всіх K ребер та описується системою:

$$f_{rip}(V_{base}, P_i) = \begin{cases} 1, & \text{якщо } (\forall k \in [1, K]: \Delta_k > 0) \vee (\forall k \in [1, K]: \Delta_k < 0) \\ 0, & \text{якщо знаки } \Delta_k \text{ відрізняються} \end{cases}. \quad (2.31)$$

Такий підхід не містить операцій ділення, реалізований без використання умовних переходів, тому є оптимізованим для виконання на CPU в умовах обробки відеопотоку в реальному часі.

2.3.2 Оцінка площі просторового перекриття

Метод трасування променя та метод псевдоскалярних добутків є швидкими. Проте при цьому вони є вразливими до крайніх випадків через неправильне положення самої точки. Наприклад, якщо транспортний засіб припаркований з порушенням і займає два суміжних паркомісця, його базова точка V_{base} може опинитися на лінії розмітки або в сусідньому слоті.

Для вирішення цієї проблеми модель доповнено розрахунком площі перекриття обмежувальної рамки b_j та паркомісця P_i .

Спочатку розраховується загальну площу багатокутника обмежувальної рамки за формулою Гаусса:

$$Area(b_j) = \frac{1}{2} \left| \sum_{k=1}^K (x_k y_{k+1} - x_{k+1} y_k) \right|. \quad (2.32)$$

Далі застосовується алгоритм відсікання багатокутників Сазерленда-Ходжмана, який послідовно відсікає частини рамки b_j , що виходять за межі паркомісця P_i , формуючи новий багатокутник перетину. Площа цього нового багатокутника позначається як S_{inter} .

Класична метрика IoU, яка використовує в знаменнику площу саме об'єднання двох фігур, є неефективною для задачі моніторингу паркомісць. Це пов'язано з тим, що транспортний засіб (наприклад, мотоцикл) може бути значно меншим за габарити самого паркувального слота, що призведе до штучного заниження показника перекриття. Тому замість площі перекриття використано площу обмежувальної рамки виявленого транспортного засобу.

Таким чином, метрика просторового перекриття слота обчислюється як відношення площі перетину до площі обмежувальної рамки автомобіля:

$$IoU_{slot} = \frac{S_{inter}}{Area(b_j)}. \quad (2.33)$$

Однак, навіть модифікована метрика має обмеження, пов'язане з геометрією перспективної проєкції. При похилому розташуванні камери обмежувальна рамка транспортного засобу включає не лише його наземну проєкцію, а й візуальну висоту кузова (дах, вікна, борти). Внаслідок цього $Area(b_j)$ суттєво перевищує реальну площу наземного контакту автомобіля, що призводить до заниження значення IoU_{slot} для великогабаритних транспортних засобів, розташованих поблизу камери.

Альтернативним підходом є використання площі паркомісця як знаменника - $S_{inter}/Area(P_i)$. Така метрика є вразливою у протилежному випадку – малогабаритний транспортний засіб (мотоцикл) на великому паркомісці може дати низьке значення перекриття навіть при повному фізичному блокуванні слота (аналогічному стандартній IoU).

Для компенсації обох випадків запропоновано використання метрики перетину відносно мінімуму, яка обчислюється за формулою:

$$IoMin = \frac{S_{inter}}{\min (Area(b_j), Area(P_i))}. \quad (2.34)$$

Така метрика автоматично обирає менший із знаменників. Якщо обмежувальна рамка значно більша за багатокутник паркомісця (великий автомобіль поблизу камери), знаменником стає $Area(P_i)$, у протилежному випадку (малий автомобіль на великому слоті) - $Area(b_j)$. Це забезпечує інваріантність метрики до співвідношення розмірів об'єкта та паркомісця, яке змінюється залежно від відстані до камери та кута огляду.

2.3.3 Формування комплексного критерію зайнятості

Об'єднуючи метод перевірки базової точки та метод розрахунку перекриття, остаточно формалізується функція просторового аналізу (2.13). Проте

застосування описаного просторового критерію виключно до окремих кадрів є вразливим до динамічних перешкод (наприклад, проїзд авто повз паркомісце). Тому отримане значення розглядається лише як миттєвий стан i -го паркомісця $\tilde{y}_{occ,i}(t)$.

Загальне правило прийняття рішення щодо статусу i -го паркомісця y_i формулюється як логічна диз'юнкція:

$$\tilde{y}_{occ,i}(t) = f_{space}(b_j, P_i) = \begin{cases} 1(\text{Зайнято}), & \text{якщо } f_{pip}(V_{base}, P_i) = 1 \vee IoMin \geq T_{iou} \\ 0(\text{Вільно}), & \text{в іншому випадку} \end{cases}, \quad (2.35)$$

де T_{iou} – заданий поріг просторового перекриття. На рис. 2.5 зображено приклад оцінки площі перекриття двох слотів авто, що зайняло 2 паркомісця.

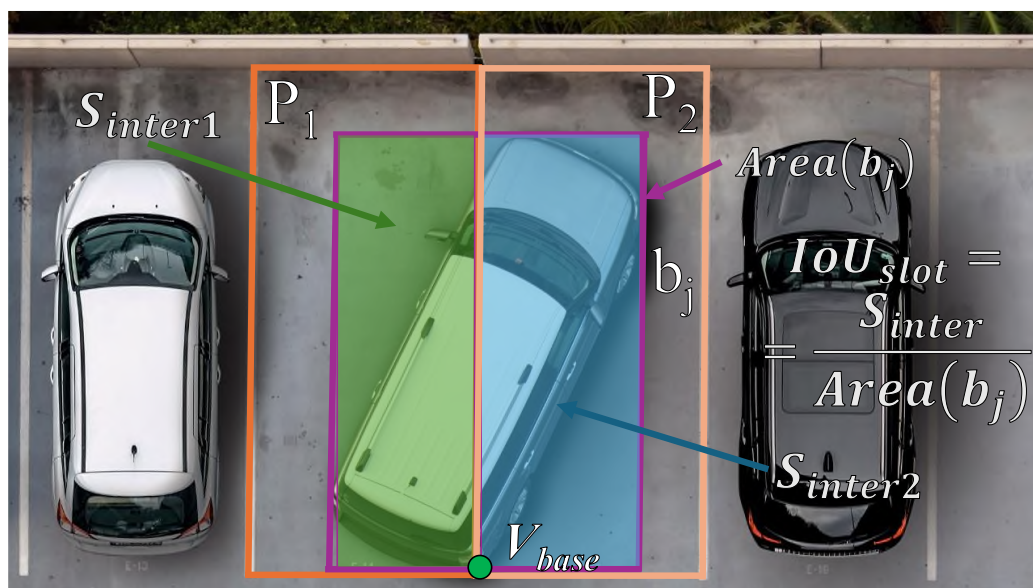


Рисунок 2.5 – Оцінка площі просторового перекриття

Після експериментального визначення емпіричним способом порогу T_{iou} , при якому слот буде вважатися зайнятим, система має позначити обидва слоти як зайняті.

Така просторова модель забезпечує надійність кіберфізичної системи при нетипових сценаріях паркування. Проте отримане значення $\tilde{y}_{occ,i}(t)$ є статичним

зрізом в певний момент часу і не враховує динаміки сцени, що вимагає подальшої часової обробки отриманих даних.

2.4 Математичне моделювання часової динаміки та фільтрації станів

Просторовий відбір дозволяє отримати миттєвий стан паркомісця $\tilde{y}_{occ,i}(t)$. Однак в умовах відеомоніторингу використання покадрових даних є вразливим до тимчасових подій. Наприклад, транспортний засіб, що рухається паркувальним майданчиком у пошуках вільного місця, буде тимчасово перетинати простір інших слотів. Геометричний алгоритм коректно зафіксує входження базової точки в полігон і змінить статус на зайнятий, однак через кілька кадрів статус знову зміниться на вільний. Таке явище називається коливанням стану.

Для забезпечення стійкості кіберфізичної системи введемо математичну модель часової фільтрації, яка базується на принципах кінцевих автоматів.

Визначимо стан паркомісця $y_{occ,i}(t)$ як остаточний, на відміну від миттєвого стану $\tilde{y}_{occ,i}(t)$.

Для зміни остаточного стану $y_{occ,i}(t)$ з одного значення в інше, миттєвий стан $\tilde{y}_{occ,i}(t)$ повинен зберігати нове значення безперервно (або з мінімальною похибкою) протягом заданого часового вікна ΔT .

Нехай відеопотік обробляється з частотою F кадрів на секунду. Тоді часовому вікну ΔT відповідає буфер розміром $N_{buf} = F\Delta T$ кадрів.

Стан системи моделюється через ковзне вікно, що зберігає історію миттєвих станів для кожного паркомісця [59]. Сума миттєвих станів у буфері за останні N_{buf} кадрів для i -го місця розраховується за формулою:

$$W_i(t) = \sum_{k=0}^{N_{buf}-1} \tilde{y}_{occ,i}(t - k). \quad (2.36)$$

Відношення (2.37) цієї суми до розміру буфера формує ймовірнісну оцінку $\rho_i(t)$ того, що паркомісце дійсно зайняте протягом даного проміжку часу.

$$\rho_i(t) = \frac{W_i(t)}{N_{buf}}, \rho_i(t) \in [0,1]. \quad (2.37)$$

Правило оновлення остаточного стану $y_{occ,i}(t)$ описується функцією (2.38) з двома порогоми – порогом підтвердження зайнятості T_{occ} та порогом підтвердження звільнення T_{free} .

$$y_{occ,i}(t) = \begin{cases} 1(\text{зайнято}), & \text{якщо } \rho_i(t) \geq T_{occ} \\ 0(\text{вільно}), & \text{якщо } \rho_i(t) \leq T_{free} \\ y_i(t-1), & \text{в іншому випадку} \end{cases}. \quad (2.38)$$

Використання двох порогів замість одного жорсткого дозволяє уникнути циклічних перемикань стану у випадках, коли ймовірнісна оцінка $\rho_i(t)$ коливається біля граничного значення (наприклад, через зовнішні чинники, які впливають на розпізнавання).

У розробленій просторово-часовій моделі фігурує ряд порогових коефіцієнтів: поріг перекриття T_{iou} , а також пороги остаточної функції зайнятості T_{occ} та T_{free} . Ці величини не є наперед заданими статичними константами, а розглядаються як конфігуровані гіперпараметри кіберфізичної системи.

Їх конкретні значення не можуть бути виведені аналітично, оскільки вони залежать від унікальних фізичних характеристик конкретної локації (висоти встановлення камери, кута огляду, фокусної відстані об'єктива). Тому задача визначення оптимальних порогів зводиться до задачі оптимізації на етапі калібрування системи.

Оптимальні значення гіперпараметрів (вектор T_{opt}) знаходяться шляхом максимізації цільової функції (2.39) якості (наприклад, метрики F1-Score (2.8)) на попередньо розміченому валідаційному наборі даних для конкретної камери.

$$T_{opt} = \arg \max_T F_1(Y_{occ,true}, Y_{occ,pred}(T)), \quad (2.39)$$

де $Y_{occ,true}$ – еталонний вектор станів паркомісць;

$Y_{occ,pred}(T)$ – вектор станів, розрахований системою при заданому наборі порогів T .

Такий підхід гарантує незалежність розробленої математичної моделі. Алгоритми залишаються незмінними та універсальними, тоді як адаптація до нових фізичних умов відеомоніторингу відбувається виключно за рахунок параметричного перекалібрування порогів.

Для наочної формалізації процесу перемикання станів системи представимо розроблений алгоритм як абстрактний детермінований скінченний автомат.

Нехай скінченний автомат задається п'ятіркою $\langle Q, \Sigma, \delta, q_0, F \rangle$, де:

$Q = \{q_0, q_1\}$ – множина можливих станів i -го паркомісця, де q_0 відповідає стану $y_{occ,i}=0$ (вільно), а q_1 відповідає стану $y_{occ,i}=1$ (зайнято);

Σ – вхідний алфавіт, що визначається безперервним спектром значень ймовірнісної оцінки $\rho_i(t) \in [0,1]$;

$\delta: Q \times \Sigma \rightarrow Q$ – функція переходів, яка визначає зміну стану системи;

$F = \{q_1\}$ – множина допускаючих станів, яка в контексті безперервного моніторингу визначає цільовий стан виявлення автомобіля.

Функція переходів δ базується на введених порогах підтвердження T_{occ} та T_{free} і описується наступним графом переходів:

1. Перехід $q_0 \rightarrow q_1$ (зайняття) відбувається тоді і тільки тоді, коли поточна оцінка досягає верхнього порогу $\rho_i(t) \geq T_{occ}$.
2. Перехід $q_1 \rightarrow q_0$ (звільнення) відбувається тоді і тільки тоді, коли поточна оцінка падає нижче нижнього порогу $\rho_i(t) \leq T_{free}$.
3. Збереження стану $q_0 \rightarrow q_0$ відбувається, поки $\rho_i(t) < T_{occ}$, стан місця залишається вільним.
4. Збереження стану $q_1 \rightarrow q_1$ відбувається, поки $\rho_i(t) > T_{free}$, стан місця залишається зайнятим.

Описана модель утворює петлю (мертву зону) в діапазоні (T_{free}, T_{occ}) , де будь-які коливання вхідного сигналу $\rho_i(t)$ ігноруються, що математично гарантує відсутність ефекту коливання стану та стабільність вихідного вектора.

Граф станів зображено на рис. 2.6.

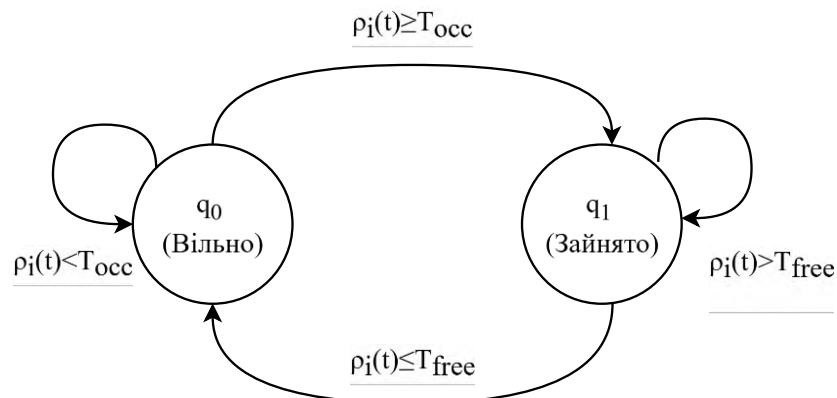


Рисунок 2.6 – Граф станів скінченного автомата системи

Впровадження цієї математичної моделі часової динаміки додає стійкості кіберфізичній системі та здатність адаптуватися до реальних фізичних процесів руху транспортних засобів, формуючи надійний фінальний вектор станів $Y_{occ}(t)$.

Для концептуального об'єднання розроблених математичних функцій у єдиний алгоритм обробки представимо процес як абстрактну структурну модель потоків даних. Ця модель ілюструє послідовність трансформацій від вхідного сигналу до кінцевого результату. На рис. 2.7 зображено цю модель.

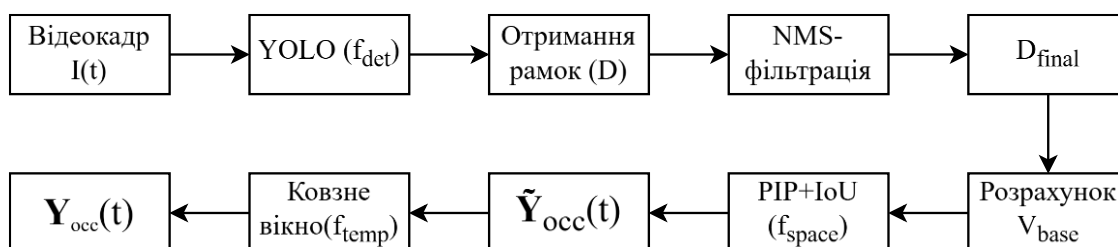


Рисунок 2.7 – Абстрактна структурна модель потоку даних

Потік даних описується наступним ланцюгом перетворень:

1. Початковий стан, який описується двовимірною матрицею пікселів поточного кадру $I(t)$.
 2. Трансформація f_{det} , суть якої полягає в нейромережевій обробці, яка генерує первинну множину розпізнавань D .
 3. Фільтрація f_{nms} , що виконує алгоритм пригнічення немаксимумів, зменшує множину D до фінальної множини унікальних об'єктів D_{final} .
 4. Просторова трансформація f_{space} , яка використовує алгоритми геометричного аналізу, що відображають множину об'єктів D_{final} та множину полігонів S на вектор миттєвих станів $\tilde{Y}_{occ}(t)$.
 5. Часова трансформація f_{temp} , у якій кінцевий автомат перетворює вектор миттєвих станів $\tilde{Y}_{occ}(t)$ на відфільтрований вектор підтверджених станів $Y_{occ}(t)$.
- Аналітично цей ланцюг можна записати як композицію функцій (2.40).

$$Y(t) = f_{temp} \left(f_{space} \left(f_{nms} \left(f_{det} (I(t)) \right) \right), S \right). \quad (2.40)$$

2.5 Висновки

У другому розділі проведено математичне моделювання процесів, що відбуваються у розроблюваній кіберфізичній системі для визначення зайнятості паркувального майданчика. Формалізовано загальну концепцію системи, яка розв'язує задачу шляхом відображення пікселів вхідного зображення на фінальний вектор логічних станів паркомісць. Концептуально процес обробки розділено на етап нейромережевої локалізації об'єктів та етап просторового аналізу. Для об'єктивної оцінки якості роботи нейромережі в умовах реального часу обрано та адаптовано стандартні метрики бінарної класифікації, зокрема Precision, Recall та Accuracy.

Розроблено математичну модель сцени паркування, яка враховує геометрію перспективної проєкції та спотворення камери. Для компенсації ефекту візуального паралаксу обґрунтовано використання опорної точки – центру нижньої грані

обмежувальної рамки. Ця точка слугує апроксимацією місця контакту транспортного засобу з площиною дороги, що дозволяє коректно позиціонувати об'єкт і уникнути обчислювально складних 3D-перетворень.

Для швидкої перевірки входження опорної точки в межі полігона паркомісця описано використання алгоритму трасування променя. Для оптимізації обчислень у випадку строго опуклих багатокутників цей алгоритм реалізовано через метод півплощин з використанням псевдоскалярного добутку векторів. Просторовий метод доповнено розрахунком площі перекриття полігонів для забезпечення надійності системи при нетипових сценаріях порушення правил паркування.

Оскільки використання лише покадрових просторових даних є вразливим до тимчасових подій, створено математичну модель часової фільтрації станів на основі абстрактного детермінованого скінченного автомата з використанням ковзного вікна. Введення ймовірнісної оцінки та використання двох порогів (для підтвердження зайнятості та звільнення) ефективно усувають проблему коливання логічного стану через динамічні перешкоди.

Всі розроблені математичні перетворення, починаючи від отримання відеокадру і закінчуючи формуванням вектора підтверджених станів, об'єднано у єдиний алгоритм обробки, який представлено у вигляді абстрактної структурної моделі потоків даних.

3 МЕТОД ПРОСТОРОВО-ЧАСОВОГО АНАЛІЗУ ДЛЯ ВИЗНАЧЕННЯ ЗАЙНЯТОСТІ ПАРКОМІСЦЬ

3.1 Загальна структура розробленого методу просторово-часової фільтрації

Розроблений метод призначений для автоматизованого визначення статусу заповненості паркувального майданчика на основі аналізу відеоданих. Суть методу полягає у комплексному просторово-часовому аналізі, що дозволяє зменшити вплив оптичних аномалій та короткочасних перекриттів місць транспортними засобами.

Загальна структура методу являє собою послідовний процес, який перетворює вхідний масив пікселів одного кадру в вектор станів паркувальних місць. Процес обробки інформації поділяється на п'ять взаємопов'язаних етапів.

Етап 1. Ініціалізація просторової моделі та отримання даних. На початковому етапі в систему завантажується математичний опис парковки – множина двовимірних статичних багатокутників $P = \{P_1, P_2, \dots, P_n\}$, кожен з яких описує геометричні межі окремого паркувального місця у площині кадру. Далі алгоритм отримує поточний кадр $I(t)$ з відеопотоку для подальшого аналізу.

Етап 2. Локалізація об'єктів. Вхідний кадр піддається аналізу за допомогою алгоритмів комп'ютерного зору для виділення об'єктів. Метою цього етапу є виявлення всіх потенційних транспортних засобів та формування для кожного з них обмежувальної рамки b_j . Результатом етапу є множина знайдених просторових рамок, незалежно від їхнього відношення до паркувальних місць.

Етап 3. Просторовий відбір та оцінка перекриття. Це ключовий етап просторового аналізу, призначений для визначення факту перебування виявленого транспортного засобу b_j в межах цільової зони P_i . Для компенсації масштабування та перспективи застосовується адаптивна метрика просторового перекриття IoMin. Результатом етапу є формування вектора миттєвих станів, де фіксується наявність або відсутність просторового збігу для кожного паркомісця у момент часу t .

Етап 4. Часова стабілізація. Оскільки миттєві просторові стани не показують реального стану в якийсь момент для водія (наприклад, проїзд автомобіля через вільне паркомісце), метод передбачає буферизацію даних. Алгоритм використовує ковзне вікно для накопичення історії станів та застосовує логіку з 2 порогами на базі скінченного автомата. Це дозволяє здійснювати зміну глобального статусу місця лише за умови тривалого збереження миттєвого стану.

Етап 5. Формування вектора станів. На фінальному етапі система генерує результуючий вектор $Y_{occ}(t) = [y_{occ,1}, y_{occ,2}, \dots, y_{occ,N}]$, де кожен елемент $y_{occ,i} \in \{0, 1\}$ відображає кінцевий статус відповідного паркувального місця на момент часу t .

Для формалізації описаної логіки та візуалізації послідовності переходів розроблено схему, яка деталізує потоки даних та умови розгалуження (рис. 3.1).

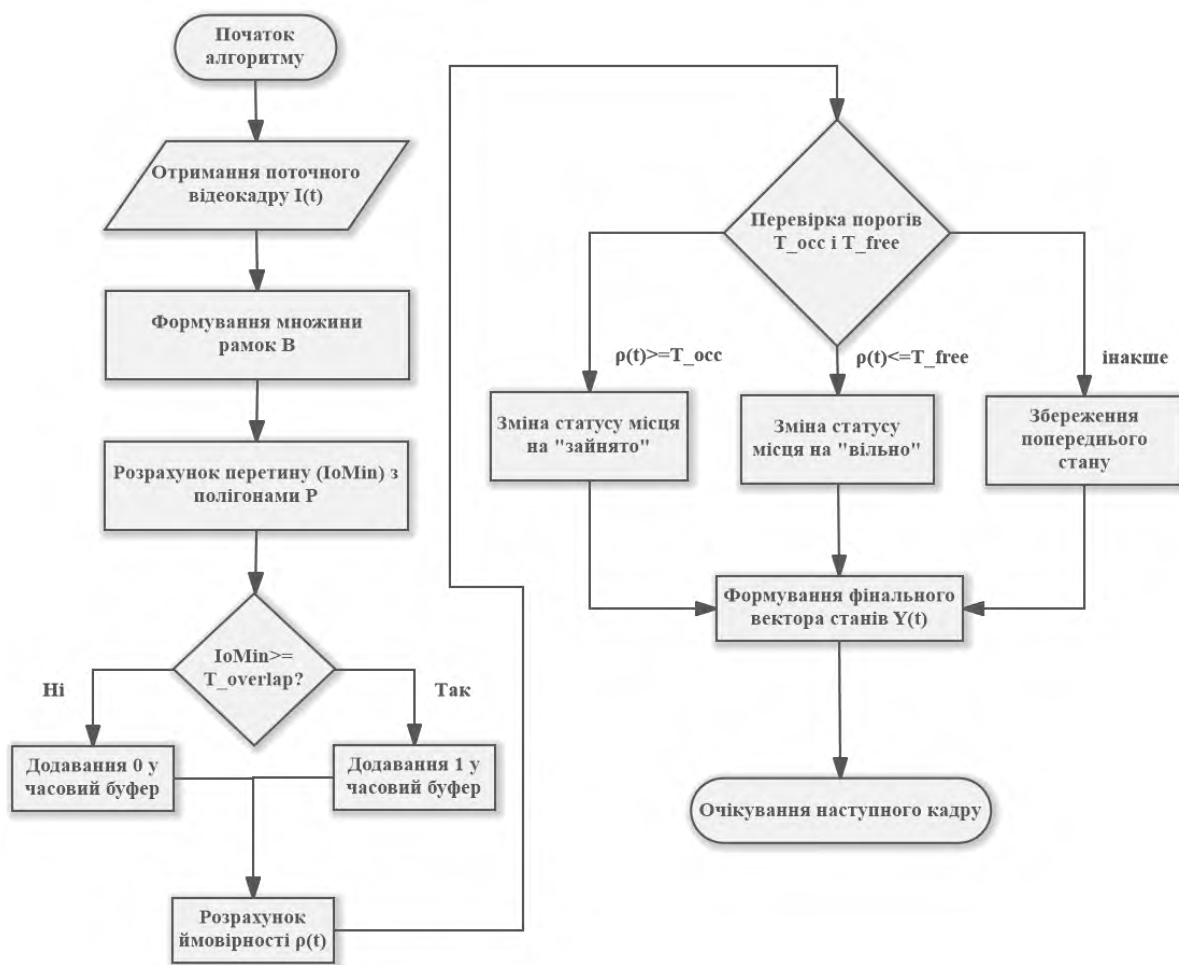


Рисунок 3.1 – схема розробленого методу просторово-часової фільтрації

3.2 Метод попередньої обробки відеопотоку та просторового маскуванню

Оригінальний відеопотік, що надходить від камери відеоспостереження, має високу надлишковість. Значна частина площі кадру містить об'єкти та зони, що не належать до зони інтересу (небо, дерева, будівлі, проїжджа частина за парковкою, паркан і т.д.).

Подача таких невідфільтрованих кадрів на вхід згорткової нейромережі призводить до використання більшої кількості ресурсів центрального процесора на аналіз зайвих об'єктів та підвищує ризик хибних розпізнавань. Для усунення цієї проблеми розроблено метод попередньої обробки та маскуванню, який виконує роль вхідного фільтра.

Процес підготовки зображення складається з чотирьох основних етапів:

1. Захоплення кадру.
2. Створення маски зони інтересу.
3. Накладання маски.
4. Передача оптимізованого зображення.

На першому етапі система здійснює зчитування поточного кадру $I(t)$ з відеопотоку. Алгоритм отримує цей кадр як матрицю пікселів, і в пам'яті фіксуються його просторові розміри для подальшої роботи.

На другому етапі алгоритм формує загальний контур майданчика. На основі заздалегідь визначених координат вершин, які описують зовнішні межі паркінгу, система будує багатокутник, який є зоною інтересу.

На базі цього багатокутника створюється двійкова матриця-маска M , розмірність якої дорівнює розмірності вхідного кадру $I(t)$. Пікселям матриці M , координати яких належать до внутрішньої області багатокутника, присвоюється значення 1, а всім зовнішнім пікселям – значення 0.

На третьому етапі алгоритм виконує операцію попиксельного множення (добуток Адамара) між матрицею кадру $I(t)$ та матрицею M . Результатом цієї дії є нове зображення, в якому вся візуальна інформація в межах паркувального майданчика залишається незмінною (оскільки множення на 1 зберігає оригінальні

значення пікселів). Водночас усі пікселі, що знаходяться поза межами зони інтересу, набувають нульового значення і зафарбовуються чорним кольором.

Створення великих однорідних областей з нулями спрощує роботу системи. Під час проходження матриці через згорткові шари нейромережі ядра фільтрів, потрапляючи на масив нулів, не намагаються знайти там об'єкти, що зменшує обчислювальну складність. Крім цього, бібліотеки лінійної алгебри оптимізують роботу з розрідженими або однорідними матрицями.

Найважливіший внесок цього етапу – це зменшення ймовірності хибних розпізнавань нейромережею автомобілів, що проїжджають вулицею або припарковані за межами цільової зони, що підвищує загальну точність методу.

На рис. 3.2 зображено приклад маскуванню кадру.



Рисунок 3.2 – Приклад маскуванню кадру

На четвертому етапі оптимізована матриця кадру передається на вхід алгоритму моделі, яка шукатиме транспортні засоби лише у потрібних межах кадру. Попередня обробка відеопотоку гарантує ефективне використання обчислювальних ресурсів, а також підвищує точність системи.

3.3 Метод фільтрації і семантичного відбору результатів розпізнавання

Результатом розпізнавання об'єктів на кадрі є формування первинної множини виявлених об'єктів D . Для забезпечення коректної роботи наступних модулів та зменшення обчислювального навантаження цю множину необхідно очистити від нерелевантних даних. Цей процес оптимізації ініціюється безпосередньо на рівні конфігурації параметрів моделі. Процес алгоритмічної фільтрації даних описується як послідовне перетворення множини і складається з трьох ключових етапів.

Етап 1. Семантичний відбір та порогове відсікання. Відповідно до математичної моделі (2.21), кожен елемент початкової множини D являє собою кортеж $d_j = (b_j, c_j, l_j)$, де b_j – геометричні координати рамки, $c_j \in [0,1]$ – оцінка впевненості моделі у тому, що в даній рамці дійсно знаходиться певний клас об'єкту, а $l_j \in L$ – ідентифікатор цього класу (L – повна множина класів, які розпізнає модель). На цьому етапі алгоритм виконує перевірку кожного кортежу d . Фільтрація здійснюється на основі одночасного виконання двох критеріїв:

- перевіряється належність розпізнаного ідентифікатора класу l_j до цільової підмножини класів $L_{target} = \{\text{"машина"}, \text{"мотоцикл"}, \text{"автобус"}, \text{"вантажівка"}\}$;

- перевіряється перевищення оцінкою впевненості c_j заданого конфігураційного порогу T_{conf} . Цей параметр емпірично встановлено на рівні $T_{conf} = 0,15$.

Якщо кортеж не задовольняє хоча б одній з цих умов, він відкидається системою як нерелевантний. Усі кортежі, що успішно пройшли фільтрацію, формують нову множину D_{conf} .

Етап 2. Пригнічення немаксимумів. Характерною алгоритмічною особливістю роботи алгоритмів комп'ютерного зору є здатність генерувати кілька обмежувальних рамок навколо одного об'єкта. Для усунення проблеми надлишковості до множини D_{conf} застосовується ітеративний алгоритм пригнічення немаксимумів. Алгоритм виконує попарне порівняння просторових координат обмежувальних рамок b_j усередині множини D_{conf} . Критерієм подібності виступає геометрична метрика взаємного перекриття IoU. Якщо алгоритм виявляє групу рамок, що перекриваються з коефіцієнтом, вищим за внутрішній допустимий ліміт, він ініціює перевірку їхніх оцінок c_j . Серед усієї групи залишається лише один кортеж із найвищим значенням впевненості c_j , тоді як усі інші дублікати вилучаються.

Етап 3. Формування фінального масиву об'єктів. У результаті виконання алгоритму формується фінальна множина D_{final} . Ця множина містить виключно унікальні кортежі транспортних засобів, які мають підтверджену високу оцінку достовірності c_j та позбавлені просторового дублювання. Множина D_{final} передається на наступний етап обробки – у розроблений модуль просторового відбору паркомісць для безпосереднього розрахунку метрики перекриття IoMin.

3.4 Метод просторового відбору транспортних засобів

Метод просторового відбору є ключовим компонентом розробленого методу, що виконує перетворення масиву просторових даних від нейромережі у вектор миттєвих станів паркувальних місць. Враховуючи високу мінливість умов моніторингу на різних майданчиках (різні кути нахилу камер, різні класи транспортних засобів і т.д.), розроблений метод підтримує декілька метрик. Така архітектура дозволяє системі мати можливість перемикатися між декількома

різними критеріями просторової оцінки або виконувати їх паралельно для збору аналітичних метрик та порівняльного тестування ефективності.

Алгоритм функціонує за наступними етапами:

1. Ініціалізація та попереднє оброблення вхідних даних.
2. Розгалуження алгоритму.
3. Визначення миттєвого стану.
4. Передача масиву станів.

На першому етапі алгоритм отримує на вхід два основні масиви даних: множину $B = \{b_1, b_2, \dots, b_M\}$, що містить координати обмежувальних рамок виявлених транспортних засобів на поточному кадрі t , та множину $P = \{P_1, P_2, \dots, P_n\}$, що описує координати статичних плоских багатокутників цільових паркомісць. Додатково система зчитує системний конфігураційний параметр, який визначає потрібний метод просторового обчислення для поточної ітерації.

На другому етапі, залежно від заданої конфігурації алгоритм спрямовує потік даних на одну з двох розгалужених гілок обробки. Аналіз виконується ітеративно для кожної пари об'єктів (b_j, P_i) .

Гілка А. Використовується метод на базі опорної координати V_{base} . Якщо система сконфігурована на використання базового геометричного підходу, алгоритм ініціює цей варіант розрахунку. Кожна обмежувальна рамка b_j перетворюється в опорну точку V_{base} , що відповідає центру нижньої грані прямокутника (точка потенційного контакту із поверхнею землі). Далі алгоритм виконує перевірку належності розрахованої точки до внутрішньої області багатокутника цільового паркомісця P_i . Ця операція виконується за алгоритмом косих добутків.

Гілка Б. Метод критерію IoMin. Якщо обрано метод метрики площі, алгоритм рахує її, а не опорну точку. Система переходить до аналізу площ. Алгоритм застосовує математичні операції відсікання багатокутників (алгоритм Сазерленда-Ходжмана) для знаходження перетину між рамкою b_j та багатокутником P_i та розраховує його площу $Area(S_{inter})$. Після обчислення простих площ обох фігур,

система обирає мінімальне значення $\min(\text{Area}(b_j), \text{Area}(P_i))$, яке йде у знаменник. Підсумкова метрика $IoMin$ розраховується як відношення площі перетину до цього знаменника. Використання функції мінімуму дозволяє алгоритму коректно обробляти об'єкти з непропорційно різними габаритами (внаслідок віддаленості від камери або впливу візуального паралаксу), що забезпечує коректність обрахунку для різних випадків.

На третьому етапі, незалежно від того, якою гілкою йшло обчислення (результат гілки А – логічна істина перевірки входження точки або гілки Б – числове значення метрики перекриття), алгоритм переходить до етапу визначення стану місця. Для гілки Б отриманий результат порівнюється з відповідним заданим порогом чутливості системи ($T_{overlap}$). Емпірично значення параметра $T_{overlap}$ визначено як 0,5. На основі цього порівняння метод формує миттєвий двійковий стан $\tilde{y}_{occ,i}(t)$ для i -го паркувального місця, де значення 1 відповідає стану зайнятості, а 0 – стану вільного місця.

На фінальному етапі алгоритм збирає обчислені індивідуальні стани у єдиний вектор миттєвих станів паркувального майданчика $\tilde{Y}_{occ}(t) = \{\tilde{y}_{occ,1}, \tilde{y}_{occ,2}, \dots, \tilde{y}_{occ,n}\}$. Оскільки ці дані є зрізом одного кадру та не враховують тимчасові перешкоди, цей вектор станів автоматично передається на вхід наступного модуля – підсистеми часової стабілізації.

3.5 Метод часової стабілізації станів

Результати роботи методу просторового відбору формують виключно статичний зріз даних для кожного ізольованого кадру. Такі миттєві стани є вразливими до тимчасового проїзду автомобілів по паркувальному майданчику, руху пішоходів, раптових змін освітлення або тимчасових перекриттів камери. Для зменшення впливу цих подій та формування достовірної інформації про статус паркувальних місць розроблено алгоритм часової стабілізації станів, що працює на базі моделі скінченного автомата з подвійним порогом перемикання. Цей метод діє

як фільтр і виконується за наступними етапами для кожного i -го паркувального місця:

1. Ініціалізація та оновлення ковзного вікна.
2. Обчислення ймовірнісної оцінки зайнятості.
3. Фільтрація станів на основі логіки скінченного автомата.
4. Формування вихідних даних.

На першому етапі алгоритм отримує на вхід вектор миттєвих двійкових станів $\tilde{y}_{occ,i}(t)$, сформований попереднім модулем в момент часу t . Для реалізації механізму пам'яті метод підтримує буфер фіксованої довжини N_{buf} для кожного паркомісця.

Під час кожної ітерації алгоритм виконує операцію зсуву: нове поточне значення $\tilde{y}_{occ,i}(t)$ записується у буфер, автоматично витісняючи найстаріше значення $\tilde{y}_{occ,i}(t - N_{buf})$. Таким чином, система постійно утримує масив останніх миттєвих станів за визначений проміжок часу.

На другому етапі, після оновлення даних у буфері, алгоритм переходить до етапу роботи з ймовірністю. Метод розраховує ймовірність зайнятості паркувального місця $\rho_i(t)$. Обчислення виконується шляхом знаходження відношення суми всіх станів (0 або 1) у буфері до його загального фіксованого розміру N_{buf} .

Якщо над вільним місцем на кілька секунд проїде інший автомобіль, він згенерує лише невелику кількість одиниць у буфері, що призведе до незначного зростання загальної ймовірності $\rho_i(t)$, недостатнього для хибного спрацювання системи.

На третьому етапі, отримана ймовірнісна оцінка $\rho_i(t)$ (зі значенням в діапазоні від 0 до 1) переводиться в чіткий нуль або одиницю. Для прийняття остаточного рішення алгоритм використовує логіку подвійного порогу. Оцінка пропускається через систему з трьома розгалуженими умовами. Якщо обчислена ймовірність досягає або перевищує верхній поріг підтвердження зайнятості ($\rho_i(t) \geq T_{occ}$), алгоритм переводить остаточний стан паркомісця $y_{occ,i}(t)$ у

значення 1 (зайнято). Якщо ймовірність падає до рівня або нижче порогу підтвердження звільнення ($\rho_i(t) \leq T_{free}$), метод класифікує місце як звільнене, переводячи стан $y_{occ,i}(t)$ у 0 (вільно). Якщо значення ймовірності перебуває в проміжку між двома порогоми $T_{free} < \rho_i(t) < T_{occ}$, то алгоритм зберігає попередній підтверджений стан паркомісця $y_{occ,i}(t - 1)$. Використання двох рознесених порогів формує мертву зону. Цей підхід усуває ефект постійного перемикавання стану місця, коли автомобіль проїжджає через це місце або коли якість кадру тимчасово погіршується.

На фінальному етапі алгоритм збирає стани всіх N паркувальних місць. Індивідуальні значення $y_{occ,i}(t)$ формуються у глобальний вектор $Y_{occ}(t) = [y_{occ,1}, y_{occ,2}, \dots, y_{occ,N}]$. Цей вектор є остаточним результатом роботи всього алгоритму. Він представляє об'єктивний стан паркувального майданчика в поточний момент часу та передається далі на вищі архітектурні рівні системи.

3.6 Оптимізація методу для систем з обмеженими ресурсами

Практичне впровадження кіберфізичних систем комп'ютерного зору вимагає використання спеціалізованих апаратних прискорювачів. У разі відсутності таких прискорювачів все навантаження йде на центральний процесор, що є менш ефективним і при цьому більш ресурсоємним. Для гарантування стабільної та автономної роботи розробленого методу просторово-часового аналізу в умовах жорстких апаратних обмежень було інтегровано три рівні алгоритмічної оптимізації.

Першою оптимізацією є фізичне обмеження ресурсів. Математичні алгоритми, що лежать в основі нейромережових моделей, схильні до глибокого розпаралелювання. У стандартному режимі виконання вони намагаються максимально використати весь ресурс процесора. А у середовищах із розподіленим доступом (наприклад, на спільних серверах) така поведінка призводить до

миттєвого перевантаження системи, вичерпання квот на системні процеси та подальшого примусового блокування алгоритму на рівні гіпервізора.

Для усунення цієї проблеми розроблений метод передбачає обмеження обчислювального потоку. Алгоритм містить внутрішні директиви, які жорстко обмежують кількість потоків для математичних операцій. Замість неконтрольованого паралельного виконання пошук об'єктів та обчислення виконуються строго послідовно, наскільки це можна забезпечити. Хоча це збільшує час обробки одного кадру, такий підхід гарантує передбачуване, низьке та рівномірне навантаження на процесор, унеможливаючи аварійне завершення роботи.

Далі оптимізація базується на аналізі динаміки предметної області. Процес паркування транспортного засобу є низькодинамічною подією, адже зміна фізичного статусу паркомісця займає десятки секунд або хвилини. Відповідно, безперервна обробка відеопотоку в режимі реального часу із частотою 25-30 кадрів на секунду є надлишковою і призводить до зайвого використання обчислювальних ресурсів.

З огляду на це, метод передбачає дискретний режим. Алгоритм здійснює періодичне вибіркове захоплення кадрів. Система зчитує один поточний кадр, повністю пропускає його через етап просторового відбору та часової фільтрації, оновлює кінцевий вектор станів, після чого ініціює паузу тривалістю в 1 секунду. Після завершення очікування цикл повторюється. Таке зниження частоти зменшує навантаження на процесор, зберігаючи при цьому достатню швидкість системи для потреб користувачів.

Остання оптимізація виконується для процесу передачі даних. Поки обчислювальний алгоритм аналізує кадри, паралельний серверний процес повинен постійно зчитувати результати для передачі їх кінцевим користувачам на вебінтерфейс. Використання класичних реляційних баз даних для організації цієї взаємодії може створити значні зайві витрати ресурсів та мати ризик взаємного блокування процесів під час одночасного доступу.

Для вирішення цієї проблеми метод використовує алгоритм атомарної заміни структурованих файлів. Механізм обміну даними працює наступним чином. Основний обчислювальний модуль форматує результуючий вектор станів $Y_{occ}(t)$ та повністю записує його у прихований файл. Після успішного завершення запису, система виконує атомарну транзакцію заміни старого робочого файлу на новий тимчасовий. Оскільки атомарна підміна виконується на рівні операційної системи за один процесорний такт, інший паралельний процес ніколи не зіткнеться з частково записаними або пошкодженими даними. Цей метод апаратно виключає виникнення конфліктів потоків, гарантуючи абсолютну цілісність даних при мінімальних витратах ресурсів.

Таким чином, дані оптимізації дозволяють створити алгоритм, здатний стабільно функціонувати на центральному процесорі спеціалізованих апаратних прискорювачів.

3.7 Формування просторових зон інтересу та еталонних даних

Для забезпечення коректного функціонування розробленого методу просторово-часової фільтрації, а також для проведення подальших експериментальних досліджень, кіберфізична система потребує первинної статичної ініціалізації.

Процес розгортання системи на новій локації стикається з проблемою відсутності універсальності, адже кожна камера відеоспостереження має унікальні параметри встановлення (висота, фокусна відстань об'єктива, кут нахилу до горизонту та азимут). Як наслідок, проєкція паркувального майданчика на площину кадру зазнає індивідуальних геометричних та перспективних спотворень.

Для вирішення цієї проблеми розроблено метод первинного конфігурування системи для окремої ділянки, яка складається з чотирьох послідовних етапів:

1. Отримання еталонного кадру.
2. Формування топології майданчика.
3. Серіалізація просторових даних.

4. Формування еталонних даних для експериментальних досліджень.

На першому етапі відбувається отримання фото майданчика. Алгоритм встановлює мережеве з'єднання з IP-камерою та здійснює захоплення одного статичного кадру. Важливою умовою цього етапу є взяття кадру за сприятливих умов видимості (відсутність атмосферних опадів, достатній рівень освітленості), а також при чистому майданчику, що забезпечує чітку ідентифікацію меж кожного паркувального місця.

На другому етапі формується топологія майданчика. Повністю автоматизувати процес розмітки не завжди можливо через зношеність дорожнього покриття, наявність снігу або часткове перекриття розмітки автомобілями. Тому методика передбачає застосування ручної розмітки людиною-оператором. Для цього використовується спеціалізований графічний інтерфейс користувача. Людина-оператор здійснює інтерактивну візуальну розмітку простору безпосередньо на еталонному кадрі, описуючи кордони кожного місця за вершинами. Алгоритм з'єднує ці вершини, формуючи замкнені опуклі багатокутники P_i кожен з яких представляє контури окремого паркувального місця. На рис. 3.3 зображено приклад такого графічного інтерфейсу.

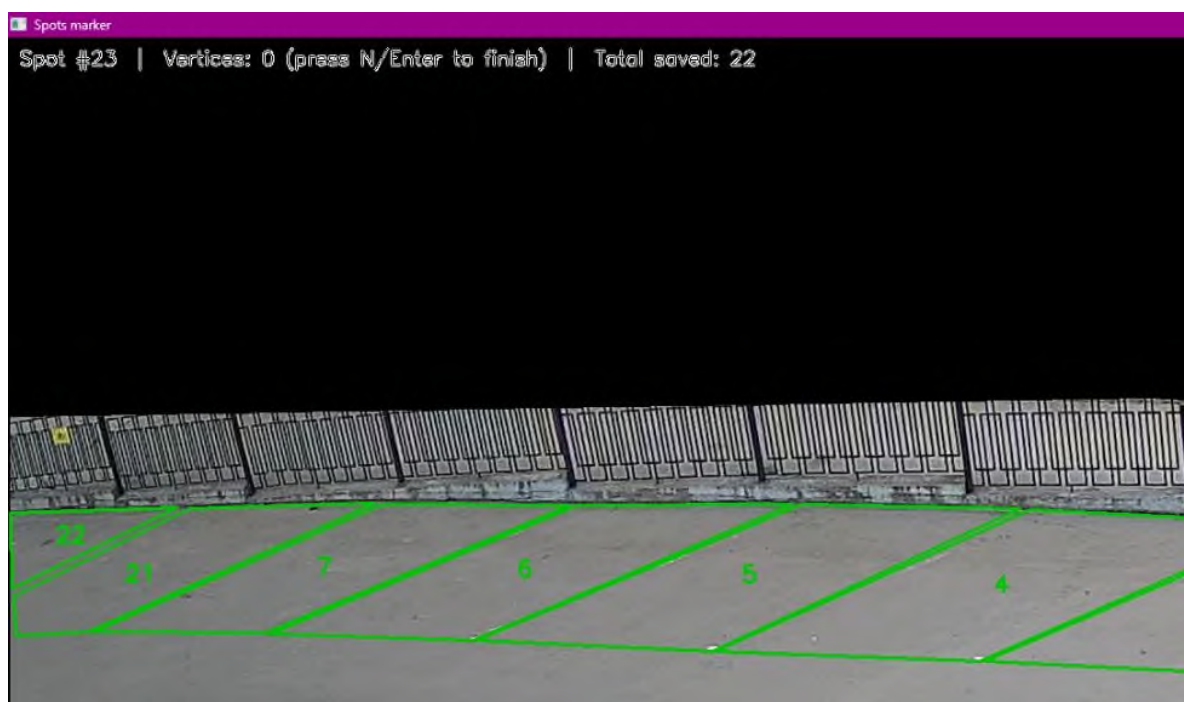


Рисунок 3.3 – Графічний інтерфейс користувача для розмітки паркомісць

На третьому етапі отримані масиви координат переводяться у текстовий формат і зберігаються у вигляді конфігураційного файлу стандарту JSON. Цей конфігураційний файл виступає джерелом вхідних даних для алгоритму під час роботи системи в режимі реального часу. Модуль просторового відбору використовує їх для розрахунку метрики IoMin.

На фінальному етапі відбувається підготовка еталонних даних для верифікації ефективності методу. Для об'єктивної оцінки точності, повноти та надійності розроблених алгоритмів необхідно мати еталон для порівняння. З цією метою методика передбачає процес ручної експертної розмітки стану майданчика на масиві тестових кадрів. Для кожного i -го паркомісця на тестовому кадрі t оператор візуально оцінює та фіксує його стан (вільно/зайнято), формуючи вектор еталонних значень $Y_{occ,gt}(t)$. Надалі цей еталон зіставлятиметься з вектором $Y_{occ}(t)$, який згенеровано системою під час експериментів. Це дозволяє розрахувати матрицю помилок та оцінити з її допомогою метрики системи. Таким чином, цей допоміжний комплекс етапів описує, як система ініціалізується даними перед її повноцінною роботою.

3.8 Комп'ютерне моделювання та експериментальна перевірка розроблених методів

Для об'єктивної оцінки ефективності розробленого методу було проведено етап комп'ютерного моделювання. Основною метою експериментів стала кількісна перевірка здатності розробленої моделі, а саме, точки V_{base} та метрики IoMin, компенсувати перспективні спотворення та здатності моделі скінченного автомата з двома порогами ефективно стабілізувати стан паркомісця. Моделювання проводилося на статичному масиві даних, зібраному в зимовий та весняний періоди, що дозволило забезпечити відтворюваність результатів та провести глибокий порівняльний аналіз.

Експеримент 1. Моделювання методів просторового відбору без часової стабілізації.

Для проведення першого етапу моделювання було сформовано контрольну вибірку кадрів. Набір даних складався з 44 незалежних кадрів, отриманих із камери відеоспостереження, встановленої під гострим кутом до горизонту. Кожен кадр містить 22 паркувальні місця, що в сумі забезпечують 968 просторових станів для аналізу. Для забезпечення об'єктивності метрик класифікації вибірка охоплює різні умови природного освітлення, а також різні ступені заповненості майданчика транспортними засобами різних габаритів. Шляхом ручної розмітки було сформовано еталонну матрицю станів, яка слугувала базисом для розрахунку матриці помилок.

У ході експерименту здійснювалося паралельне моделювання та порівняння чотирьох підходів до просторового відбору:

- базовий метод (M_1), який перевіряє входження геометричного центру обмежувальної рамки транспортного засобу в координати багатокутника паркомісця;
- метод опорної точки V_{base} , який перевіряє входження центру нижньої грані обмежувальної рамки;
- метод IoU_{slot} , який розраховує відношення площі перетину обмежувальної рамки і паркомісця до площі обмежувальної рамки з порогом $T=0,25$;
- метод $IoMin$, який розраховує відношення площі перетину фігур до мінімальної площі серед них з порогом $T=0,5$.

Кількісні результати роботи кожного з алгоритмів (метрики класифікації Precision, Recall та Accuracy) наведені у таблиці 3.1.

Аналіз отриманих результатів моделювання дозволив виявити критичні вразливості методів.

Базовий метод через центр точки коректно визначає випадки, де автомобіль справді припарковано на конкретному паркомісці (95,98% чіткості). Проте у випадках, коли через паралакс точка виходить з меж одного паркомісця і потрапляє на інше, метод дає хибнонегативний результат для цього місця (80,93% повноти).

Таблиця 3.1 – Метрики роботи 4 методів просторового відбору

Метод	Precision	Recall	Accuracy
M1 (центр)	0,9598	0,8093	0,8698
V_{base}	0,9122	0,4260	0,6436
IoU_{slot} (T=0,25)	0,9021	0,9037	0,8874
IoMin (T=0,50)	0,9147	0,8984	0,8926

Метод опорної точки V_{base} , незважаючи на очікуваний успіх, продемонстрував дуже низьку ефективність. Показник Recall розпізнавання становить критично малі 42,6%. При цьому Precision методу – 91,2%. Проте метод пропустив 57,4% автомобілів на паркомісцях. Аналіз хибнонегативних спрацювань цього методу показав, що причиною помилок методу є недооцінювання впливу кута нахилу камери. Через сильний нахил оптичної осі камери та перспективне масштабування обмежувальна рамка автомобіля формується таким чином, що її нижня грань з V_{base} геометрично виходить за межі цільового багатокутника, навіть якщо автомобіль фізично ідеально припаркований у його межах. Така поведінка зображена на рис. 3.4.

Для нижнього ряду авто (який ближче до камери) нижня грань чітко потрапляє у межі паркомісця. А у верхньому ряді (який далі від камери, відповідно, об'єкти під більшим кутом) обмежувальні рамки правильно припаркованих авто виходять за межі, і таким чином точка V_{base} не потрапляє у паркомісце.

Недоліки виявлено і при моделюванні метрики IoU_{slot} . Оскільки перспектива збільшує площу обмежувальної рамки для об'єктів на передньому плані, знаменник формули стрімко зростає. Це призводить до зниження метрики: її значення падає нижче порогу, і алгоритм ігнорує зайняте місце.

Натомість результати моделювання методу IoMin довели його стійкість до оптичних спотворень. Заміна знаменника на функцію мінімуму з двох площ нівелювала ефект розширення обмежувальних рамок. Метод продемонстрував чіткість на рівні 91,47% та повноту 89,84%. Метод виявився дещо краще ніж IoU_{slot} і це пояснюється тим, що у випадках де через велику площу самої обмежувальної

рамки на фоні площі перетину, значення IoU_{slot} не вистачає для подолання порогу, а у цей час $IoMin$ вирішує цю проблему через взяття площі самого паркомісця і тим самим значення цієї метрики досягає потрібного порогу.



Рисунок 3.4 – Візуалізація роботи метрики V_{base} (фіолетові точки)

Необхідно відзначити, що частка хибнонегативних помилок (8,6%) здебільшого пояснюється не математичними обмеженнями самої метрики $IoMin$, а нездатністю нейромережі сформувати обмежувальну рамку для автомобілів у темний час доби. Це спостерігається і для інших методів, так як вони також не отримують цих даних. Також обидва методи $IoMin$ та IoU_{slot} завдяки їхній природі роботи з площинами автоматично вирішують проблему, коли два або більше авто зливаються в один об'єкт для нейромережі. Так як формули оперують площинами перетину в чисельнику і площинами окремих об'єктів в знаменнику, то закономірно, сусідні місця позначаються як зайняті ($IoMin$ при цьому працює краще).

Крім цього, варто розуміти, що ці отримані значення є миттєвими станами ($\tilde{Y}_{occ}(t)$), тому тимчасова зміна стану паркомісця не вплине на глобальний стан, якщо це не відбувається протягом тривалого часу.

Згідно табл. 3.1 базовий метод має більшу чіткість, ніж метод IoMin. Проте за рахунок різниці у повності точність методу IoMin вища, ніж у базового. Враховуючи, що різниця у чіткості є невеликою, а зростання повноти при цьому є суттєвим, метод IoMin виявився найкращим серед усіх протестованих.

Експеримент 2. Моделювання часової стабілізації.

Отримання високих показників точності на статичних ізольованих кадрах не гарантує стабільної роботи кіберфізичної системи у реальному часі через наявність об'єктів, що тимчасово можуть вплинути на стан паркомісця. Для перевірки методу часової стабілізації було обрано динамічну сцену тривалістю 27 послідовних кадрів. Сценарій відображав процес маневрування та паркування великогабаритного автомобіля, траєкторія руху якого візуально перекривала декілька суміжних вільних паркувальних місць.

Під час цього експерименту порівнювалися базовий метод, метод IoMin та IoU_{slot}. Спочатку змодельовану сцену було оброблено методами без використання часового модуля. Як і передбачалося, такий підхід згенерував значну кількість хибнопозитивних спрацювань. У моменти, коли автомобіль проїжджав над вільними місцями, система одразу змінювала статус відповідних місць.

Далі отриманий масив миттєвих станів був пропущений через модуль часової стабілізації. Для експерименту було сконфігуровано параметри кільцевого буфера: глибина ковзного вікна $N_{buf} = 5$ та пороги спрацювання $T_{occ} = 0,6$ і $T_{free} = 0,3$. Порівняльні результати ефективності системи до та після застосування часової стабілізації наведено у табл. 3.2.

Аналіз даних підтверджує, що інтеграція часової фільтрації сильно змінює поведінку системи у кращу сторону. Це дозволило досягти чіткості на рівні 100% для базового методу та IoMin. Ковзне вікно успішно спрацювало як фільтр, поглинувши короточасні (тривалістю 1–2 кадри) перекриття вільних місць маневруючим автомобілем. Показники $\rho_i(t)$ для паркомісць не змогли накопичити

достатнє значення для подолання верхнього порогу T_{occ} . Таким чином, хибні спрацювання були усунуті.

Таблиця 3.2 – Результати ефективності системи

Метод	Precision	Recall	Accuracy
M1 без фільтрації	0,9964	0,8900	0,9411
M1 + часовий фільтр	1,0000	0,8900	0,9428
IoU _{slot} без фільтрації	0,9058	0,9029	0,9007
IoU _{slot} + часовий фільтр	0,9103	0,8867	0,8956
IoMin без фільтрації	0,9896	0,9256	0,9562
IoMin + часовий фільтр	1,0000	0,9126	0,9545

Водночас моделювання зафіксувало незначне зниження показника повноти для методу IoMin з 0,9256 (до фільтрації) до 0,9126 (після фільтрації). Ця зміна не є дефектом алгоритму, а відображає поведінку скінченного автомата. Ковзний фільтр має затримку реакції. Система навмисно утримує статус паркомісця незмінним протягом кількох кадрів, очікуючи остаточного підтвердження стабільності нового стану. Це гарантує надійність даних, що передаються користувачам, і є прийнятним компромісом у задачах моніторингу. Також варто відзначити, що розмітка еталонних даних проводилася у ручному режимі, тому сам факт зайнятості місця на кадрі може бути різним залежно від експерта.

На рис. 3.5 зображено колаж з 3 кадрів для демонстрації роботи алгоритму.

Автомобіль рухається заднім ходом до кінця території парковки. На першому кадрі автомобіль тимчасово займає 17-те паркомісце. Так як значення $\rho_i(t)$ не доходить до T_{occ} , то місце вважається вільним. На наступному кадрі 17-те місце доволі довго перебуває у зайнятому стані, значення $\rho_i(t)$ доходить до T_{occ} і місце позначається як зайняте. На третьому кадрі, коли більша частина автомобіля залишила це паркомісце, значення метрики спадає і, з часом, $\rho_i(t)$ знижується до рівня T_{free} і місце стає вільним.

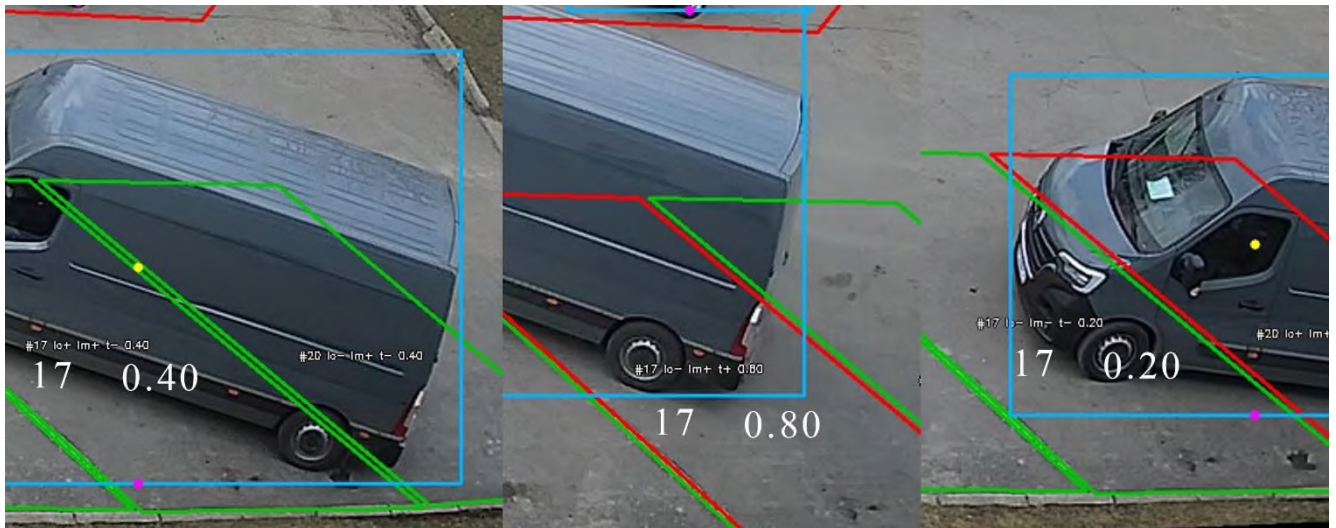


Рисунок 3.5 – Візуалізація роботи часового фільтру

Варто зазначити, що на останньому кадрі 17-те паркувальне місце все ще частково закрито транспортним засобом, але система класифікує його як вільне. Ця поведінка суворо регулюється емпірично відкаліброваним порогом T_{overlap} , що використовується в розрахунку $IoMin$ та IoU_{slot} . В рамках експерименту T_{overlap} визначено як 0,5. Вибір його значення є компромісом. Зниження T_{overlap} підвищує б чутливість системи до часткових перекриттів, але одночасно збільшує ризик помилкових спрацьовувань, спричинених проїжджаючим транспортом. Тому T_{overlap} є ключовим параметром налаштування, який необхідно адаптувати до конкретної геометричної конфігурації та бізнес-логіки цільового паркінгу.

Результати комп'ютерного моделювання підтверджують життєздатність та ефективність розроблених методів. Застосування просторового аналізу метрикою $IoMin$ та часового фільтра дозволяє створити надійний алгоритм, що функціонує в умовах оптичних спотворень та динамічних завад.

3.9 Висновки

У третьому розділі розроблено метод просторово-часового аналізу для визначення зайнятості паркомісць у кіберфізичній системі моніторингу

паркування. Запропонований підхід формує алгоритм, який перетворює результати нейромережевого розпізнавання на вектор станів паркомісць.

Описано алгоритм попередньої обробки та маскуванню, що дозволяє відсікати нерелевантні зони кадру ще до етапу розпізнавання. Формалізовано процес семантичного відбору та очищення результатів локалізації, що спирається на порогове відсікання та пригнічення немаксимумів.

Для усунення проблеми коливання станів, викликані тимчасовими перешкодами, розроблено метод часової стабілізації на базі скінченного автомата. Використання ковзного вікна та логіки подвійного порогу забезпечує ефективне визначення кінцевого стану паркомісця.

З метою забезпечення стабільного функціонування методів на пристроях з обмеженими обчислювальними ресурсами запропоновано комплекс алгоритмічних оптимізацій, зокрема фізичне обмеження обчислювального потоку, зменшення частоти обробки до мінімально прийнятної та атомарний запис векторів станів без використання БД. Описано загальну методику формування багатокутників паркомісць та збору еталонних даних для об'єктивного оцінювання системи.

Проведене комп'ютерне моделювання розроблених методів експериментально підтвердило їхню ефективність. Зокрема, результати моделювання довели, що метрика IoMin збільшує повноту розпізнавання з 42,6 % (у порівнянні з методом опорної точки) до 89,8 %, при цьому зберігаючи високий рівень чіткості, а інтеграція часового фільтра дозволяє досягти чіткості класифікації на рівні 100 %.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ

4.1 Архітектура та стек технологій програмного забезпечення

На основі математичної моделі та методу просторово-часового аналізу було здійснено проєктування та розробку програмного забезпечення кіберфізичної системи моніторингу паркувального простору. Основним підходом до архітектурного проєктування обрано концепцію апаратної незалежності. Це передбачає створення програмного комплексу, здатного ефективно функціонувати на базі будь-яких обчислювальних пристроїв без додаткового обладнання. Така архітектура виключає необхідність використання спеціалізованих обчислювальних кластерів або дорогих дискретних графічних прискорювачів, спираючись виключно на ресурси центрального процесора. Для забезпечення високого рівня портативності, кросплатформності та ізольованого розгортання системи використовуються віртуальні середовища та мова програмування Python. Це дозволяє легко масштабувати рішення та переносити його між різними машинами у разі необхідності.

Програмна архітектура системи розділена на два незалежні компоненти.

Головним компонентом є обчислювальне ядро, реалізоване у вигляді фонових процесу-демона. Цей модуль інкапсулює всю обчислювальну логіку розробленого методу. Процес обробки розпочинається із захоплення RTSP-потoku з IP-камери. Підсистема захоплення відеоданих реалізована без постійного утримання мережевого з'єднання. Замість цього застосовано алгоритм перервного отримання кадру, що складається з етапів встановлення сесії, миттєвого захоплення кадру та негайного розриву з'єднання. Це запобігає переповненню внутрішніх буферів програмного декодера кадрів та виключає проблему накопичення часової затримки, яка виникає під час декодування великої кількості кадрів.

Отриманий кадр передається на модуль попередньої обробки та після цього на модуль фільтрації і відбору результатів розпізнавання, який реалізовано з

використанням полегшеної моделі архітектури YOLOv8n з використанням фреймворку Ultralytics та бібліотеки комп'ютерного зору OpenCV.

Наступний етап просторового аналізу виконується за допомогою реалізованих алгоритмів обчислювальної геометрії, описаних у математичній моделі.

На фінальному етапі отримані просторові дані пропускаються через модуль часової фільтрації, перетворюючи миттєвий вектор станів на кінцевий вектор фактичних станів місць паркувального майданчика.

Клієнтський рівень системи реалізовано на базі мікрофреймворку Flask. На рис. 4.1 зображено схему архітектури програмної системи.

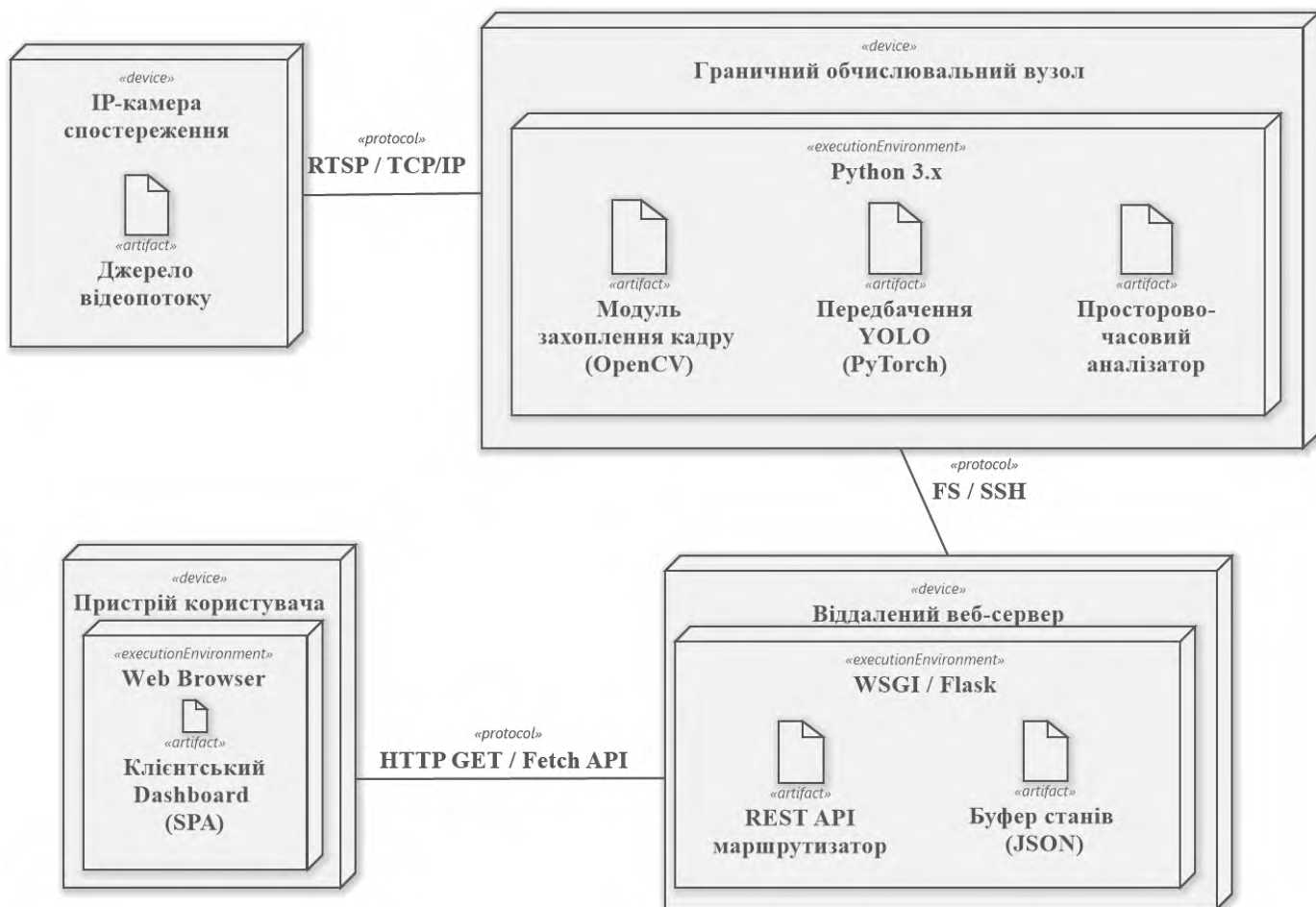


Рисунок 4.1 – Схема архітектури програмної системи

Під час вибору засобу міжпроцесної взаємодії між обчислювальним ядром та веб-інтерфейсом користувача було розглянуто два основні варіанти – реляційна

база даних або взаємодія файлом. Для збільшення швидкодії системи, а також для збереження простоти її розгортання та надійності роботи було обрано файлову взаємодію.

Використання баз даних створює зайве навантаження на дискову підсистему та має ризик блокування таблиць при одночасному доступі до даних з двох процесів на одній системі. А використання файлової системи за допомогою атомарних операцій на рівні операційної системи працює швидше і не має такого ризику. Обчислювальне ядро після завершення кожної ітерації перетворює вектор станів паркувальних місць у структурований JSON-об'єкт, а кадр з паркувального майндачика з цими даними комбінується у анотоване фото. Ці об'єкти записуються у тимчасові файли у пам'яті. Після повного збереження та підтвердження цілісності інформації система ініціює системні виклики атомарної заміни (реалізовані за допомогою методу `os.replace`). Цей метод гарантує, що операція заміщення старого файлу новим відбувається миттєво. Таким чином, модуль веб-інтерфейсу під час читання файлу ніколи не стикається з пошкодженими, порожніми або наполовину записаними даними.

При рознесенні програмних компонентів системи на 2 різних ізольованих середовища виконання використовується заміна через SSH (системна команда `scp`), а логіка при цьому залишається та ж: спочатку віддалено передається тимчасовий файл, а потім через SSH віддалено ініціюється команда заміни `mv`. Перед цим необхідно обмінятися ключами для доступу по SSH між цими системами.

Для забезпечення стандартизації інформаційного обміну між процесами було розроблено уніфіковану структуру формату передачі даних. Інформація містить як загальну системну статистику, так і деталізований вектор станів кожного окремого паркувального місця. Базова структура об'єкта передачі даних має наступний вигляд:

```

{ "timestamp": "2026-04-21 16:59:35", "total": 22,
  "free": 18, "occupied": 4, "method": "iomin",
  "spots": { "1": {"occupied": false, "rho": 0.0}, "2":
    {"occupied": false, "rho": 0.0}, ...} }

```

Така ієрархічна структура мінімізує обсяг корисного навантаження під час зчитування файлу клієнтським застосунком та дозволяє виконувати десеріалізацію вбудованими засобами мови програмування без залучення сторонніх бібліотек-парсерів.

Другим незалежним компонентом розробленої архітектури є модуль веб-інтерфейсу користувача, завданням якого є надання доступу до результатів моніторингу через звичайний веб-браузер.

Серверна частина цього модуля побудована на базі легкого мікрофреймворку Flask. Вибір цього інструменту зумовлений його мінімалізмом, відсутністю зайвих вбудованих залежностей та високою продуктивністю при виконанні завдань маршрутизації.

Архітектура клієнт-серверної взаємодії побудована на принципі асинхронного опитування. Клієнтський скрипт використовує інтерфейс Fetch API для фонового надсилання запитів до сервера із заданою періодичністю (3 секунди). Отримавши JSON-об'єкт з вектором станів, скрипт оновлює усі дані про паркувальний майндачик на сторінці. Таким чином змінюються колірні індикатори на схемі паркінгу, оновлюються цифрові лічильники вільного простору та завантажуються нові анотовані кадри фоново для користувача, зберігаючи при цьому плавну роботу без необхідності повного перезавантаження веб-сторінки.

4.2 Алгоритм роботи обчислювального ядра та розробка інтерфейсу користувача

Логіку роботи обчислювального модуля побудовано за принципом безперервного циклу, який забезпечує поетапну обробку кадрів. Архітектуру цього модуля розділено на три відокремлені зони для мінімізації зв'язності між програмними компонентами та спрощення подальшої підтримки коду.

Першою складовою архітектури є зона даних та пам'яті, яка відповідає за комунікацію із зовнішнім апаратним середовищем та операційною системою. Однією із її задач є мережева взаємодія з камерою відеоспостереження для

отримання поточного кадру за протоколом потокової передачі RTSP. Також цей блок відповідає за запис масивів даних на фізичний носій сервера. Процес збереження даних налаштований описаним вище чином: використовуючи тимчасовий текстовий файл і процедуру атомарної заміни.

Другою складовою виступає зона ядра, яка містить бізнес-логіку системи та функціонує незалежно від специфіки сторонніх бібліотек. На цьому рівні реалізовано алгоритми розрахунку розроблених метрик просторового перекриття між знайденими транспортними засобами та заданими багатокутниками паркувальних місць. Також тут зосереджено механізми ініціалізації, зсуву та очищення кільцевих буферів, на основі яких працює часовий фільтр, що перетворює миттєві стани на масив остаточних станів паркомісць на основі математичної моделі скінченного автомата з подвійним порогом спрацювання.

Третьою складовою визначено зону зовнішніх бібліотек і фреймворків, куди делеговано виконання найбільш ресурсоємних розрахунків, які не відбуваються у ядрі. До цієї зони маршрутизуються виклики до бібліотеки OpenCV, яка відповідає за декодування кадрів і роботу з ними, та PyTorch, яка відповідає за ініціювання роботи моделі YOLOv8n.

Алгоритм захоплення працює таким чином, що обчислювальний процес встановлює мережеве з'єднання виключно у момент необхідності отримання нового кадру. Після встановлення сесії програма зчитує декілька перших кадрів потоку, так як це необхідно для коректної ініціалізації декодера. Далі захоплюється один кадр для аналізу, після чого з'єднання примусово розривається. Завершивши повний цикл обробки, процес переходить у режим очікування тривалістю одну секунду. Це необхідно для чіткої швидкості функціонування системи на стабільному рівні одного кадру за секунду, запобігання накопиченню мережевої затримки та зменшення навантаження на ЦП.

Взаємодія зон відповідальності зображена на рис. 4.2.

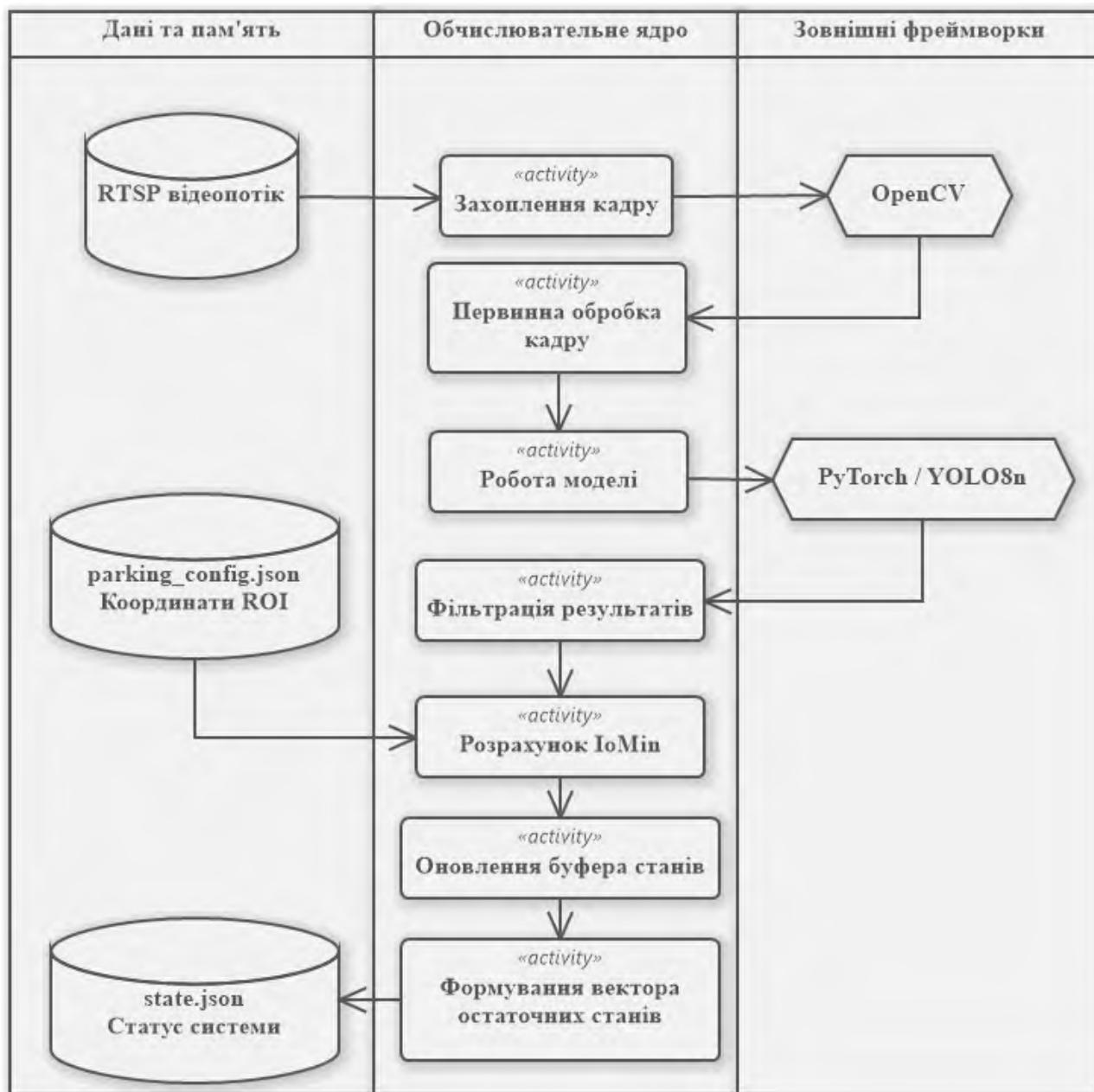


Рисунок 4.2 – Процес обробки кадру з розділеними зонами відповідальності

З метою додаткового зниження навантаження на ЦП розроблено алгоритм масштабування просторових координат. Для тестової камери конфігураційний файл, який містить координати зон інтересу, сформовано з основного відеопотоку з високою роздільною здатністю (2560 на 1440 пікселів). Аналіз зображень такого розміру вимагає великих витрат оперативної пам'яті та процесорного часу. З огляду на це, система налаштована на отримання даних із додаткового потоку камери, який апаратно передає зображення зі зниженою роздільною здатністю 640 на 360 пікселів.

Для забезпечення геометричної сумісності наявних конфігураційних даних із новим зменшеним відеопотоком реалізовано алгоритм масштабування. Після зчитування параметрів ширини та висоти вхідного зображення система виконує пропорційний перерахунок координат усіх визначених вершин багатокутників паркувальних місць згідно з обчисленими коефіцієнтами масштабування. Метрики просторового перекриття на основі площ достатньо стійкі до масштабування, що дозволяє виконувати розрахунки на зменшених зображеннях з мінімальними втратами точності класифікації об'єктів (в межах 1%).

З метою обмеження використовуваних програмою ресурсів процесора під час тривалої безперервної роботи середовища зміни для важких бібліотек встановлені так, щоб обмежити їх виконання до одного обчислювального потоку.

Для забезпечення взаємодії системи з оператором паркувального майданчика клієнтську частину розроблено у вигляді інформаційної панелі, яка візуалізує результати роботи обчислювального ядра. Графічний інтерфейс розділено на два функціональні блоки: область візуалізації даних та інформаційно-статистичну панель. На рис 4.3. наведено фрагмент інтерфейсу візуалізації даних.

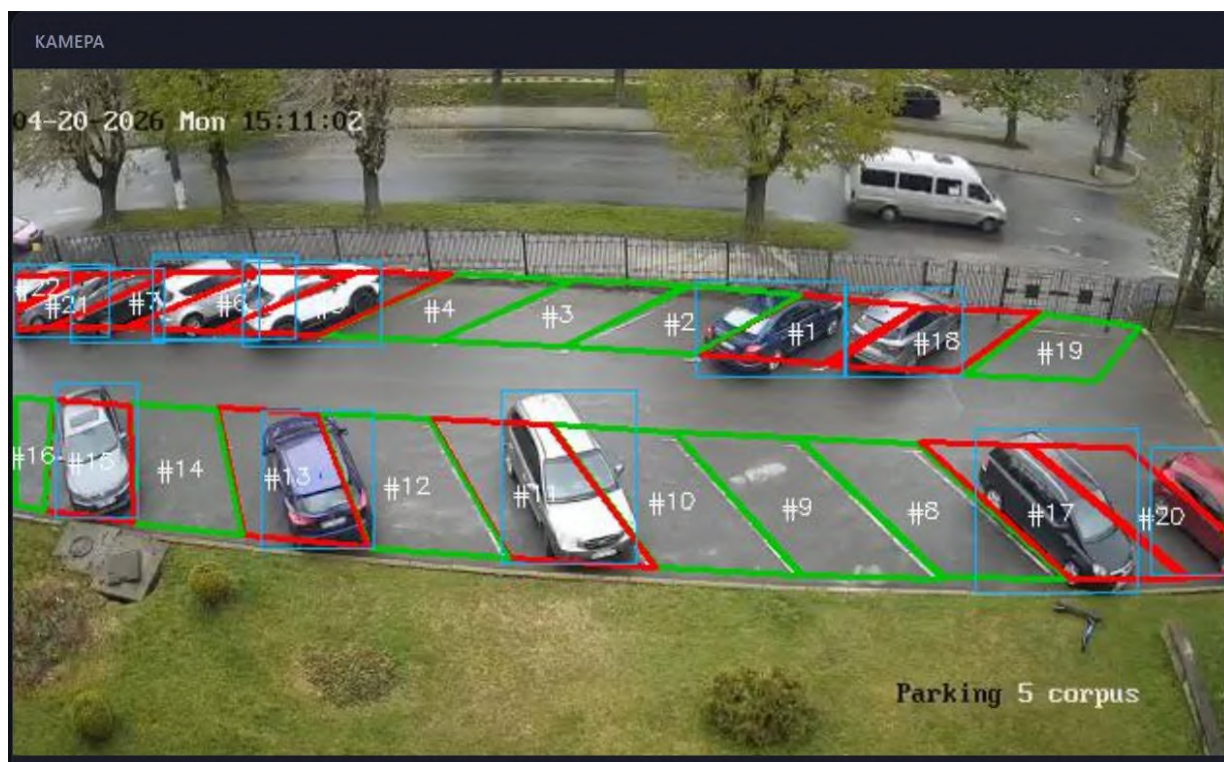


Рисунок 4.3 – Анотований кадр з відеокамери

Після завершення аналізу кадру результат роботи ядра є анотованим кадром без маски. Статус кожного окремого паркувального місця позначається безпосередньо поверх кадру. Вільні паркувальні місця позначаються зеленим кольором, тоді як зайняті місця перефарбовуються у червоний колір. Також на кадрі позначено номери паркомісць для їх ідентифікації згідно з конфігураційним файлом, а також обмежувальні рамки автомобілів для кращого розуміння того, що на даному кадрі бачить система.

Особливістю розробленого графічного інтерфейсу є відмова від повного перемальовування об'єктної моделі документа під час кожного асинхронного опитування сервера. Клієнтський скрипт порівнює отриманий вектор станів із попередньо збереженим у локальній пам'яті. Зміна статусів в інтерфейсі ініціюється виключно для тих паркувальних місць, статус яких фактично змінився порівняно з попереднім тактом опитування. Анотований кадр замінюється шляхом оновлення джерела фото самим браузером. Цей алгоритм оновлення знижує навантаження на підсистему рендерингу браузера, дозволяючи системі плавно функціонувати на малопотужних пристроях.

Другий функціональний блок інтерфейсу відповідає за виведення статистики. Ця панель у режимі реального часу відображає загальну кількість паркувальних місць, динамічні лічильники вільних та зайнятих слотів, а також інформацію по кожному місцю разом з часом останнього оновлення і активним методом обчислення зайнятості.

Реалізацію клієнтської частини виконано мовою JavaScript без використання сторонніх фреймворків. Логіку оновлення даних побудовано на використанні вбудованих інструментів для виконання асинхронних мережевих запитів (Fetch API), за допомогою яких браузер періодично звертається до сервера для отримання актуального файлу станів інфраструктури та анотованого кадру.

Зміна статистичних даних, як і зміна кадру, відбувається асинхронно на стороні клієнта, що дозволяє відображати актуальну інформацію без необхідності повного перезавантаження сторінки. Вебсервер при цьому зосереджений виключно на маршрутизації вхідних мережевих запитів і працює як REST API.

На рис. 4.4 зображено цей блок інтерфейсу.

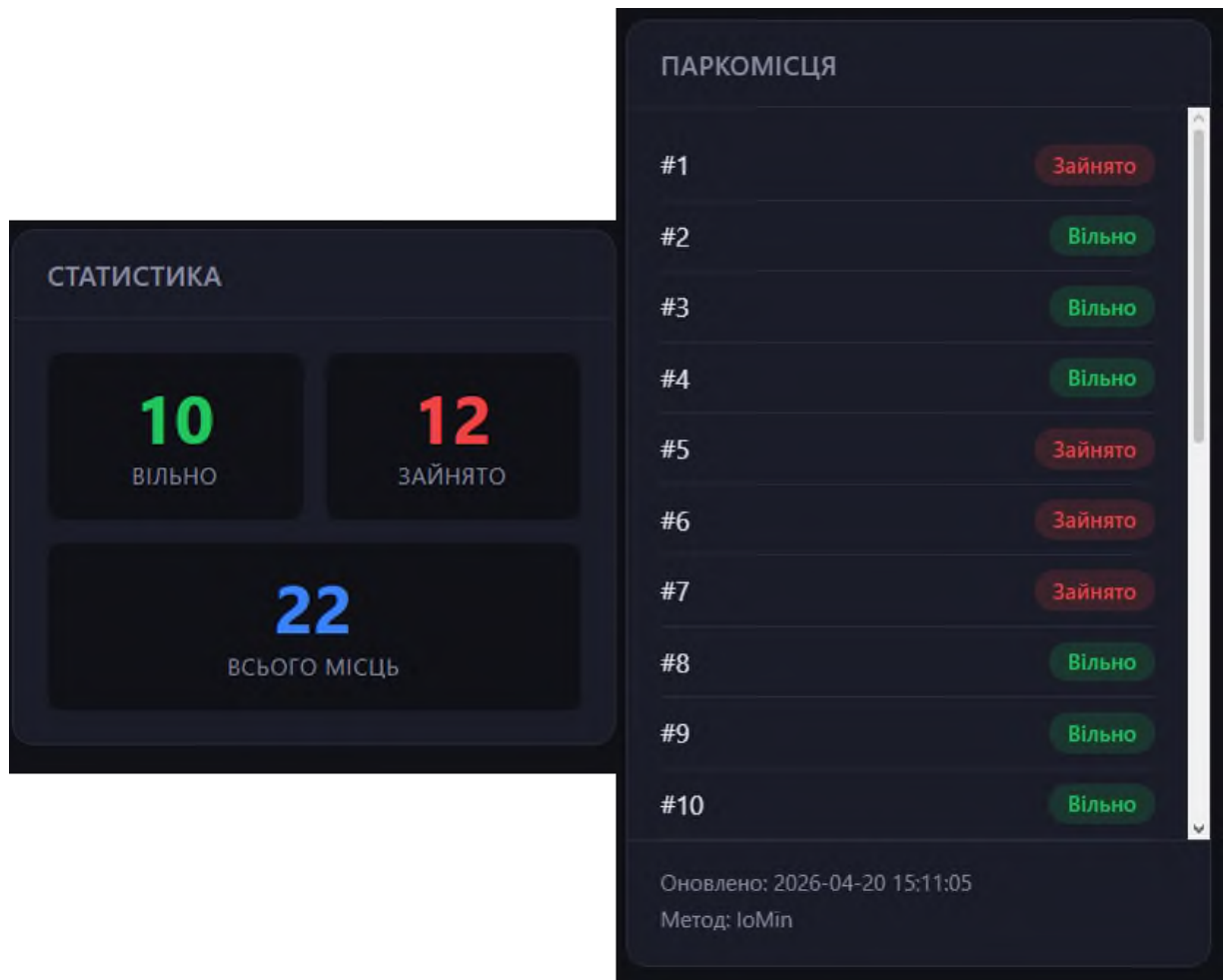


Рисунок 4.4 – Загальна і детальна статистики майданчика

4.3 Експериментальна оцінка продуктивності та поетапна оптимізація ресурсної ефективності системи

Для забезпечення коректного проведення експериментальних досліджень та первинного налаштування кіберфізичної системи було розроблено допоміжне програмне забезпечення. З метою підготовки тестової вибірки реалізовано алгоритм автоматизованого захоплення відеокадрів, у який інтегровано механізм уникнення повтору шляхом порівняння хешів з попереднім кадром. Цей механізм дозволив автоматично відсікати візуально ідентичні зображення, що зменшило обсяг збережених даних і збільшило варіативність. Для конфігурування топології

паркувального майданчика створено утиліти з графічним інтерфейсом користувача. Зазначені інструменти надали можливість розмітити координати вершин багатокутників для кожного паркувального місця на еталонному кадрі для генерації необхідних конфігураційних файлів для всіх наступних етапів тестування.

Перший етап експериментальних досліджень проводився на локальному обчислювальному стенді, оснащеному сучасним восьмиядерним центральним процесором AMD Ryzen 7 7700. Головною метою цього етапу було визначення метрик апаратного споживання системою. Для комплексного аналізу було проведено перехресне тестування чотирьох конфігурацій обробки: аналіз основного відеопотоку високої роздільної здатності та додаткового, кожен з яких почергово оброблявся у базовому багатопотоковому режимі та в оптимізованому режимі з одним обчислювальним потоком. Результати тестування наведено у табл. 4.1.

Таблиця 4.1 – Порівняння споживання ресурсів на локальному обчислювальному стенді

Тест	Режим потоку	Обмеження процесора	Завантаження CPU (сер.), %	ОЗП (сер.), МБ	Час обробки кадру (сер.), мс
1	Основний 2560×1440	1 потік	8,12	414,7	22,30
2	Основний 2560×1440	Багатопотоковий	8,30	417,3	21,51
3	Вторинний 640×360	1 потік	6,12	354,1	22,35
4	Вторинний 640×360	Багатопотоковий	6,08	356,5	22,41

Отримані результати локального тестування виявили архітектурну закономірність. Абсолютний час безпосередньої обробки кадру нейромережею залишився фактично незмінним у всіх чотирьох конфігураціях. Це пояснюється тим, що архітектура моделі виконує внутрішнє масштабування вхідного тензора до фіксованого розміру перед початком математичних обчислень. Однак застосування режиму обмеження потоків та переведення алгоритму на обробку зменшеного додаткового потоку дозволили знизити навантаження на центральний процесор під час процедури декодування відеосигналу, а також зменшити обсяг споживання оперативної пам'яті комп'ютера. Майже відсутня зміна у споживанні ЦП при переході від однопоточкового до багатопоточкового режиму пояснюється обчислювальною потужністю ЦП, при якій у режимі без обмежень вистачає ресурсів, еквівалентних однопоточковому режиму. Графік навантаження ЦП у 4 тестах зображено на рис. 4.5.

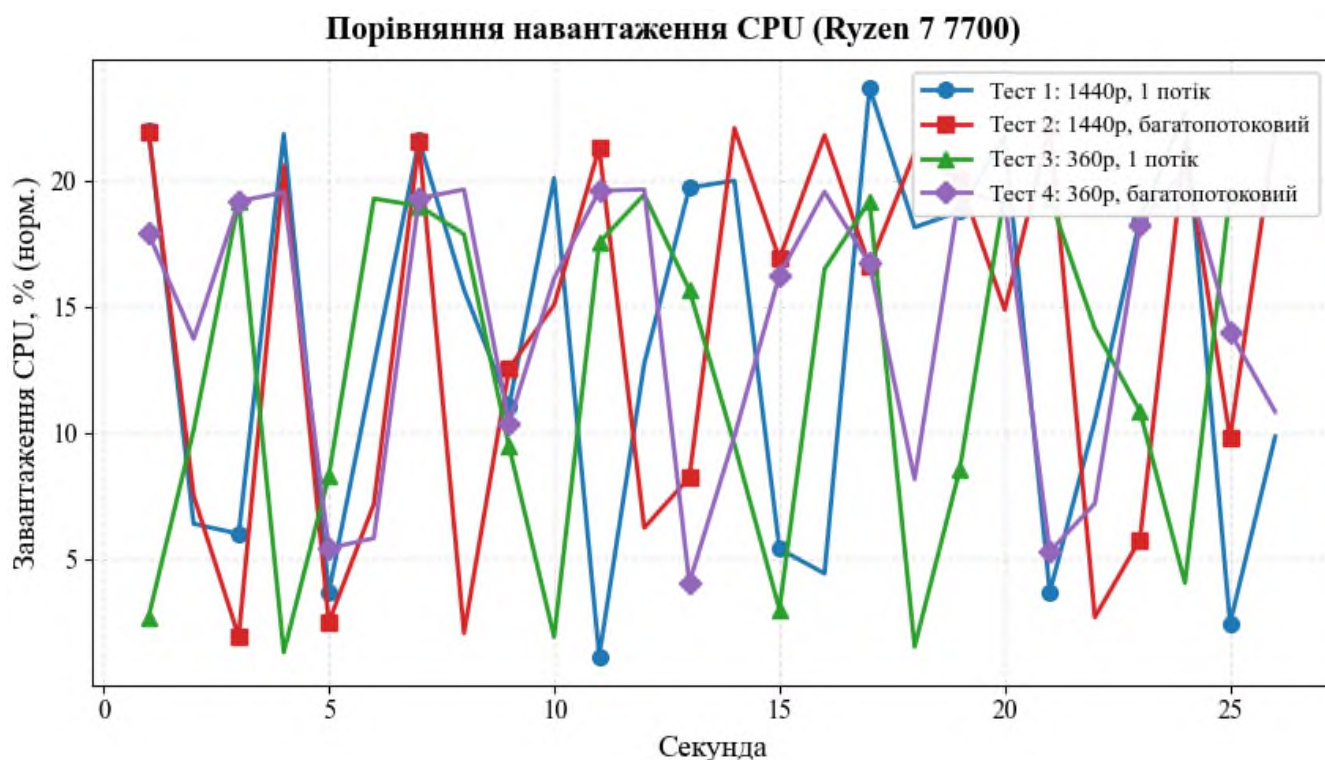


Рисунок 4.5 – Порівняння навантаження ЦП у 4 тестах

Другий етап експериментів полягав у перенесенні обчислювального ядра у цільове робоче середовище – на віртуалізований серверний хостинг із жорсткими

квотами на процесорне споживання, оперативну пам'ять і кількість процесів. Початковий запуск неоптимізованого програмного коду, який передбачав роботу з основним потоком високої роздільної здатності у режимі реального часу, виявив проблеми.

Неоптимізований процес постійно споживав 125% виділеної потужності, що є лімітом квоти, а також регулярно перевищував кількість доступних процесів. Неможливість отримання необхідних апаратних ресурсів призводила до систематичного відставання алгоритму обробки від реального часу на тридцять і більше секунд. Таке відставання спричинило переповнення внутрішнього буфера відеодекодера, внаслідок чого на вхід нейромережі почали надходити пошкоджені та візуально деформовані кадри. Було виявлено артефакти стиснення, порушення кольоропередачі та пересвічені ділянки, що знижували загальну точність розпізнавання транспортних засобів. На рис. 4.6 зображено виведення команди `top` для неоптимізованого та оптимізованого варіантів відповідно.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4003154	*****	20	0	3786712	738476	138380	R	124.2	1.1	4:34.53	python3.11_bin
3900583	*****	20	0	26752	3760	2952	S	0.0	0.0	0:00.02	bash
3937128	*****	20	0	26752	3636	2864	S	0.0	0.0	0:00.01	bash
3937821	*****	20	0	65700	3988	3376	R	0.0	0.0	0:00.87	top
4041061	*****	20	0	272768	31344	10816	S	0.0	0.0	0:00.46	python3.11_bin
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
532453	*****	20	0	2975272	413660	137712	S	57.5	0.6	0:51.72	python3.11_bin
3937821	*****	20	0	65700	3944	3332	R	0.3	0.0	0:04.22	top
3900583	*****	20	0	26752	3760	2952	S	0.0	0.0	0:00.02	bash
3937128	*****	20	0	26752	3620	2848	S	0.0	0.0	0:00.01	bash
4041061	*****	20	0	272768	31848	10792	S	0.0	0.0	0:01.21	python3.11_bin

Рисунок 4.6 – Виведення команди `top`

Для розв'язання проблеми буферизації та стабілізації процесу було розроблено алгоритм дискретного підключення до відеокамери. Цей алгоритм передбачає наступний цикл: встановлення мережевого з'єднання, скидання накопиченого буфера, захоплення лише одного актуального кадру та відключення від камери. Математичні бібліотеки були програмно обмежені одним

обчислювальним потоком, а алгоритм переведено на використання додаткового потоку зниженої роздільної здатності. Це дозволило зменшити затримку в часі до прийнятних 10 секунд в середньому та забезпечило обробку коректних кадрів. При цьому навантаження на процесор сервера знизилося до діапазону від сорока до шістдесяти відсотків, а споживання оперативної пам'яті скоротилося до чотирьохсот мегабайтів.

Різниця у споживанні неоптимізованої версії програми 700 МБ ОЗП на сервері і лише близько 400 МБ на локальному ПК пояснюється різницею версій програми. На сервері у момент знімка top працювала старіша реалізація – з тривалим з'єднанням і накопиченням кадрів у буферах, тоді як на локальному ПК виміри проводилися вже після переходу на дискретне підключення.

Фінальним етапом експериментальної оцінки стало добове стрес-тестування повністю оптимізованої версії кіберфізичної системи у серверному середовищі. Протягом двадцяти чотирьох годин безперервний моніторинг показників здійснювався засобами вбудованої панелі керування ресурсами сервера. На рис.4.7 зображено графік завантаження ЦП під час тесту.



Рисунок 4.7 – Графік завантаження ЦП під час добового тесту

Короточасні досягнення верхніх лімітів завантаження процесора (що відображаються синім пунктиром на графіку) відбуваються в середньому 3–6 разів

на добу. Отже, розроблена система здатна протягом повної доби функціонувати в умовах обмежених обчислювальних ресурсів без використання дискретних графічних прискорювачів.

4.4 Забезпечення відмовостійкості та алгоритми обробки виняткових ситуацій

Кіберфізична система вимагає автономності та здатності функціонувати в безперервному цілодобовому режимі без втручання людини. Під час роботи у реальних умовах виникають різноманітні апаратні збої, перебої в електропостачанні та програмні помилки. Архітектура програмного забезпечення повинна обов'язково містити вбудовані механізми самовідновлення та забезпечення високого рівня відмовостійкості.

Одним із найбільш вразливих вузлів розробленої системи є мережева взаємодія з джерелом відеоданих. Під час тестової експлуатації було реалізовано алгоритм обробки втрати зв'язку з відеокамерою. Бібліотеки комп'ютерного зору генерують критичну помилку читання, внаслідок чого миттєво завершується основний обчислювальний процес. Для запобігання такій поведінці було впроваджено програмний механізм перехоплення виняткових ситуацій. Замість аварійної зупинки алгоритм переходить у режим очікування, здійснюючи періодичні спроби відновлення мережевого з'єднання із заданим часовим інтервалом. Усі події, пов'язані з втратою та відновленням зв'язку, паралельно записуються у файл журналу подій, що дозволяє провести аналіз та вирішити ці проблеми у майбутньому .

Наступним етапом стало забезпечення безперервності роботи процесу на рівні операційної системи. Програмні та апаратні середовища схильні до непередбачуваних збоїв (наприклад, через нестачу ресурсів, помилки сегментації пам'яті або зовнішнє примусове завершення).

Для здійснення автоматичного перезапуску фонових процесів у випадку його непередбачуваного критичного завершення було задіяно системний планувальник

завдань операційної системи. Планувальник налаштовано на періодичну (5хв) ініціалізацію наступного командного скрипта, який бере на себе функції зовнішнього наглядача за станом системи:

```
/bin/bash -lc 'if ! ps aux | grep -E "[p]ython3.11_bin.*[c]ore.py" >
/dev/null; then cd /home/$USERNAME$/mastery_parking && nohup
/home/$USERNAME$/virtualenv/mastery_parking/3.11/bin/python core.py >>
core.log 2>&1 & fi'
```

Цей скрипт перевіряє, чи є вже запущений процес ядра інтерпретатором Python 3.11. Якщо ні, то процес переходить у папку проєкту для збереження відносної позиції запуску і запускає скрипт в режимі без переривань системою з перенесенням виведення у файл і виведенням помилок у фоновому режимі.

4.5 Висновки

У четвертому розділі на основі розроблених математичних моделей та методу було створено програмне забезпечення кіберфізичної системи. Архітектуру програмного комплексу побудовано з урахуванням необхідності розгортання системи без використання дискретних графічних прискорювачів на будь-якій стандартній системі. Структурно систему розділено на два незалежні процеси: обчислювальне ядро та інтерфейсний модуль. Використання механізму атомарної заміни файлів для міжпроцесної взаємодії дозволило відмовитися від застосування ресурсоємних реляційних баз даних.

Під час реалізації програмних засобів було впроваджено розділення зон відповідальності між серверною та клієнтською частинами. Перенесення операцій графічного відмальовування станів паркувальних місць на сторону браузера дозволило знизити загальне обчислювальне навантаження на сервер. Завдяки використанню технології асинхронних запитів забезпечено динамічне оновлення інформаційної панелі без необхідності перезавантаження вебсторінки.

Локальне тестування апаратного споживання засвідчило сталість часу обробки кадру нейромережею незалежно від використаного потоку камери. Розгортання системи у віртуалізованому серверному середовищі із жорсткими

квотами вимагало впровадження додаткових оптимізацій. Досліджено, що застосування моделі дискретного підключення до відеопотоку, алгоритмічне обмеження математичних обчислень одним потоком та перехід на обробку додаткового потоку зі зниженою роздільною здатністю дозволили усунути проблему переповнення відеобуфера, збільшити стабільність системи та зменшити споживання ресурсів.

Впроваджено механізми для забезпечення високого рівня відмовостійкості програмного забезпечення. Використання системного планувальника завдань у поєднанні з алгоритмами обробки виняткових ситуацій підвищило загальну стабільність системи при виникненні збоїв.

За результатами проведеного добового стрес-тестування доведено, що розроблений програмний комплекс споживає в межах від сорока до шістдесяти відсотків виділеного логічного ядра центрального процесора. Отримані експериментальні дані підтверджують готовність створеної кіберфізичної системи до цілодобової безперебійної експлуатації в умовах обмежених ресурсів граничних обчислювальних вузлів.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень вдосконалено метод просторово-часового аналізу для визначення зайнятості паркомісць. Також набула подальшого розвитку апаратно-незалежна кіберфізична система візуального моніторингу паркувального простору.

Поставлену мету досягнуто шляхом розв'язання таких основних завдань:

- проаналізовано сучасний стан, існуючі апаратно-програмні комплекси та методи комп'ютерного зору у сфері візуального моніторингу паркувального простору;
- розроблено математичні моделі просторового відбору об'єктів та часової стабілізації станів для компенсації ефекту візуального паралаксу;
- удосконалено метод просторово-часового аналізу відеоданих шляхом впровадження адаптивної метрики просторового перекриття та алгоритмів фільтрації на базі скінченного автомата з подвійним порогом;
- спроектовано та розроблено програмне забезпечення кіберфізичної системи з урахуванням архітектурної вимоги щодо виконання нейромережевого прогнозування на центральному процесорі;
- проведено експериментальні дослідження для оцінки точності розпізнавання та ресурсної ефективності оптимізованої системи в реальних умовах експлуатації.

Завдяки впровадженню розробленої метрики просторового перекриття вдалося компенсувати перспективні спотворення зображення, що дозволило досягти показника повноти розпізнавання 89,8 %. Застосування методу часової стабілізації на основі скінченного автомата дозволило нівелювати вплив динамічних перешкод від маневруючого транспорту, що усунуло хибні коливання станів та підвищило чіткість класифікації до 100 %.

Удосконалено архітектуру програмного забезпечення кіберфізичної системи візуального моніторингу. Впровадження моделі дискретного підключення до відеопотоку та перехід на використання додаткового потоку зі зниженою

роздільною здатністю дозволили усунути проблему переповнення відеобуфера. Експериментально підтверджено, що застосовані оптимізації зменшили споживання оперативної пам'яті до 300-400 мегабайтів.

Впровадження результатів роботи дозволило створити економічно ефективний програмний комплекс, який здатний безперебійно функціонувати виключно на потужностях центрального процесора, споживаючи лише 40-60% виділеного логічного ядра. Це підтверджує придатність системи до цілодобової експлуатації в умовах обмежених обчислювальних ресурсів вузлів без необхідності використання дискретних графічних прискорювачів.

За темою кваліфікаційної роботи опубліковано одну наукову статтю [60] в електронному фаховому науковому виданні «Кібербезпека: освіта, наука, техніка» (2025. – Вип. 1 (29). – С. 536–547).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. IOT Parking: Все про розумні системи паркування | Kyivstar Business Hub. Kyivstar Business Hub – корпоративний блог для бізнесу. URL: <https://hub.kyivstar.ua/articles/rozumni-systemy-parkuvannya-tendencziyi-ta-mozhlyvosti-dlya-ukrayinskyh-mist> (дата звернення: 04.03.2026).
2. What is the hub and spoke model?. Transvirtual. URL: <https://www.transvirtual.com/blog/what-is-the-hub-and-spoke-model/> (date of access: 04.03.2026).
3. Automatic parking systems | SEA Company. Компанія SEA | Виробництво електроніки, електронні компоненти, електротехніка | Компанія SEA. URL: <https://www.sea.com.ua/en/automatic-parking-systems/> (date of access: 04.03.2026).
4. Perz I. How SKIDATA's Mobility Suite Makes Urban Parking Management Simple. Parking Network. URL: <https://www.parking.net/parking-news/skidata-gmbh/mobility-suite-for-urban-parking-management> (date of access: 04.03.2026).
5. Tiba parking systems. TIBA Parking Systems. URL: <https://tibaparking.com/> (date of access: 04.03.2026).
6. У столиці запустили новий застосунок міських сервісів «Київ Цифровий». Офіційний портал КМДА - Головна. URL: https://kyivcity.gov.ua/news/u_stolitsi_zapustili_noviy_zastosunok_miskikh_servisiv_kiv_tsifroviy/ (дата звернення: 04.03.2026).
7. У додатку Kyiv Smart City тепер можна сплатити за паркування погодинно. Офіційний портал КМДА - Головна. URL: https://kyivcity.gov.ua/news/u_dodatku_Kyiv_Smart_City_teper_mozhna_splatiti_za_parkuvannya_pogodinno/ (дата звернення: 04.03.2026).
8. Parking - Welcome to Buntaiste Technologies. Welcome to Buntaiste Technologies. URL: <https://buntaiste.com/parking/> (date of access: 04.03.2026).
9. IoT Parking Sensors by Nwave. NWAIVE. URL: <https://www.nwave.io/smart-parking-sensor/> (date of access: 04.03.2026).

10. LLC P. ParkMobile: Park. Pay. Go. - Apps on Google Play. Android Apps on Google Play. URL: <https://play.google.com/store/apps/details?id=net.sharewire.parkmobilev2> (date of access: 04.03.2026).
11. PayByPhone | Simplifying your journey. PayByPhone | Simplifying your journey. URL: <https://www.paybyphone.com/> (date of access: 04.03.2026).
12. How Smart Parking Sensors Work: The Technology Behind Efficient Cities. Viziosense. URL: <https://www.viziosense.com/post/how-smart-parking-sensors-work-the-technology-behind-efficient-cities> (date of access: 04.03.2026).
13. ANPR / parking lot management | Hikvision. Hikvision. URL: <https://www.hikvision.net.my/parkinglotmanagement> (date of access: 04.03.2026).
14. Parklio Parking ANPR | All-in-one Solution. Parklio. URL: <https://parklio.com/en/parking-solutions/anpr> (date of access: 04.03.2026).
15. Autlo - clever parking solutions. Autlo. URL: <https://autlo.com/> (date of access: 04.03.2026).
16. Модуль розпізнавання номерних знаків ТЗ | Компанія СЕА. Компанія СЕА | Виробництво електроніки, електронні компоненти, електротехніка | Компанія СЕА. URL: https://www.sea.com.ua/ua/parkovochnoe_oborudovanie/modul-sea-anpr/ (дата звернення: 04.03.2026).
17. Keytop One-stop Parking Solutions | www.ikeytop.com | China. KEYTOP Smart Parking. URL: <https://www.ikeytop.com/> (date of access: 04.03.2026).
18. Hikvision Global - Leading the future of AIoT. URL: <https://www.hikvision.com/content/dam/hikvision/uk/marketing-portal/brochures/solutions/Smart%20Parking%20Area%20Solution.pdf> (дата звернення: 04.03.2026).
19. У Києві стартував пілот із впровадження «розумної» системи паркування. Офіційний портал КМДА - Головна. URL: https://kyivcity.gov.ua/news/u_kiyevi_startuvav_pilot_iz_vprovadzhennya_rozumno_si_stemi_parkuvannya/ (дата звернення: 04.03.2026).

20. Comprehensive Review on Vehicle Detection, Classification and Counting on Highways / P. Premaratne et al. *Neurocomputing*. 2023. P. 126627. URL: <https://doi.org/10.1016/j.neucom.2023.126627> (date of access: 04.03.2026).
21. MmWave Radar and Vision Fusion for Object Detection in Autonomous Driving: A Review / Z. Wei et al. *Sensors*. 2022. Vol. 22, no. 7. P. 2542. URL: <https://doi.org/10.3390/s22072542> (date of access: 04.03.2026).
22. Vehicle Detection Algorithms for Autonomous Driving: A Review / L. Liang et al. *Sensors*. 2024. Vol. 24, no. 10. P. 3088. URL: <https://doi.org/10.3390/s24103088> (date of access: 04.03.2026).
23. From Handcrafted to Deep Features for Pedestrian Detection: A Survey / J. Cao et al. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021. P. 1. URL: <https://doi.org/10.1109/tpami.2021.3076733> (date of access: 05.03.2026).
24. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks / K. Zhang et al. *IEEE Signal Processing Letters*. 2016. Vol. 23, no. 10. P. 1499–1503. URL: <https://doi.org/10.1109/lsp.2016.2603342> (date of access: 05.03.2026).
25. Deep Learning Techniques for Vehicle Detection and Classification from Images/Videos: A Survey / M. A. Berwo et al. *Sensors*. 2023. Vol. 23, no. 10. P. 4832. URL: <https://doi.org/10.3390/s23104832> (date of access: 05.03.2026).
26. Local binary features for texture classification: Taxonomy and experimental study / L. Liu et al. *Pattern Recognition*. 2017. Vol. 62. P. 135–160. URL: <https://doi.org/10.1016/j.patcog.2016.08.032> (date of access: 05.03.2026).
27. Ali A. M., Eltarhouni W. I., Bozed K. A. On-Road Vehicle Detection using Support Vector Machine and Decision Tree Classifications. *ICEMIS'20: The 6th International Conference on Engineering & MIS 2020*, Almaty Kazakhstan. New York, NY, USA, 2020. URL: <https://doi.org/10.1145/3410352.3410803> (date of access: 05.03.2026).
28. Vehicle detection for autonomous parking using a Soft-Cascade AdaBoost classifier / A. Broggi et al. *2014 IEEE Intelligent Vehicles Symposium (IV)*, MI, USA, 8–

11 June 2014. 2014. URL: <https://doi.org/10.1109/ivs.2014.6856490> (date of access: 05.03.2026).

29. Marcomini L. A., Cunha A. L. A Comparison between Background Modelling Methods for Vehicle Segmentation in Highway Traffic Videos. 2018. URL: <https://arxiv.org/abs/1810.02835>.

30. A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection / H. Wang et al. *IEEE Intelligent Transportation Systems Magazine*. 2019. Vol. 11, no. 2. P. 82–95. URL: <https://doi.org/10.1109/mits.2019.2903518> (date of access: 05.03.2026).

31. Mask R-CNN / K. He et al. *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 22–29 October 2017. 2017. URL: <https://doi.org/10.1109/iccv.2017.322> (date of access: 05.03.2026).

32. Fusion of Deep Sort and Yolov5 for Effective Vehicle Detection and Tracking Scheme in Real-Time Traffic Management Sustainable System / S. Kumar et al. *Sustainability*. 2023. Vol. 15, no. 24. P. 16869. URL: <https://doi.org/10.3390/su152416869> (date of access: 05.03.2026).

33. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / S. Ren et al. *Proceedings of the Advances in Neural Information Processing Systems*. Montréal, QC, Canada, 7. P. 91–99.

34. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon et al. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016. 2016. URL: <https://doi.org/10.1109/cvpr.2016.91> (date of access: 05.03.2026).

35. Faster R-CNN: Towards real-time object detection with region proposal networks / S. Ren et al. *Advances in Neural Information Processing Systems*. 2015. Vol. 28. P. 1137–1149.

36. Wang Y., Karimi H. Advanced deep learning models and algorithms for spatial-temporal data. *Big Data*. 2024. P. 227–253.

37. Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 21–26

July 2017. 2017. URL: <https://doi.org/10.1109/cvpr.2017.690> (date of access: 05.03.2026).

38. Wen H., Dai F. A Study of YOLO Algorithm for Multi-target Detection. *J. Adv. Artif. Life Robot.* 2021. Vol. 2. P. 70–73.

39. Song X., Gu W. Multi-objective real-time vehicle detection method based on yolov5. *2021 International Symposium on Artificial Intelligence and its Application on Media (ISAIAM)*, Xi'an, China, 21–23 May 2021. 2021. URL: <https://doi.org/10.1109/isaiam53259.2021.00037> (date of access: 05.03.2026).

40. Hussain Jumah Z., M. Chyaid A. A Review of Deep Learning Approaches for Vehicle Classification and Tracking in Diverse Conditions. *Journal of Al-Qadisiyah for Computer Science and Mathematics.* 2025. Vol. 17, no. 2. URL: <https://doi.org/10.29304/jqscsm.2025.17.22196> (date of access: 05.03.2026).

41. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation / R. Girshick et al. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, 23–28 June 2014. 2014. URL: <https://doi.org/10.1109/cvpr.2014.81> (date of access: 05.03.2026).

42. Liu W., others. SSD: Single Shot MultiBox Detector. *Computer Vision – ECCV 2016* / ed. by B. Leibe et al. Cham, 2016. Vol. 9905. URL: https://doi.org/10.1007/978-3-319-46448-0_2.

43. Real-Time Vehicle Detection Based on Improved YOLO v5 / Y. Zhang et al. *Sustainability.* 2022. Vol. 14, no. 19. P. 12274. URL: <https://doi.org/10.3390/su141912274> (date of access: 05.03.2026).

44. Real-Time Flying Object Detection with YOLOv8 / D. Reis et al. 2023. URL: <https://doi.org/10.48550/arXiv.2305.09972>.

45. Khanam R., Hussain M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv.* 2024. URL: <https://doi.org/10.48550/arXiv.2410.17725>.

46. Wang C.-Y., Bochkovskiy A., Liao H.-Y. M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC,

- Canada, 17–24 June 2023. 2023.
 URL: <https://doi.org/10.1109/cvpr52729.2023.00721> (date of access: 05.03.2026).
47. Feature Pyramid Networks for Object Detection / T.-Y. Lin et al. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 21–26 July 2017. 2017. URL: <https://doi.org/10.1109/cvpr.2017.106> (date of access: 05.03.2026).
48. Path Aggregation Network for Instance Segmentation / S. Liu et al. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, 18–23 June 2018. 2018. URL: <https://doi.org/10.1109/cvpr.2018.00913> (date of access: 05.03.2026).
49. Popereshnyak S., Chornobryvets D. Real-time parking space monitoring system based on computer vision. *Modern Machine Learning Technologies*. 2025.
50. Vehicle Occurrence-Based Parking Space Detection / P. R. L. de Almeida et al. *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Honolulu, Oahu, HI, USA, 1–4 October 2023. 2023. URL: <https://doi.org/10.1109/smc53992.2023.10394316> (date of access: 05.03.2026).
51. Smart parking with pixel-wise ROI selection for vehicle detection using YOLOv8, YOLOv9, YOLOv10, and YOLOv11 / G. P. C. P. da Luz et al. *Internet of Things*. 2026. Vol. 36. P. 101858. URL: <https://doi.org/10.1016/j.iot.2025.101858> (date of access: 05.03.2026).
52. Coleiro A., Scerri D., Briffa I. Car parking detection in a typical village core street using public camera feeds. *2020 IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin)*, Berlin, Germany, 9–11 November 2020. 2020. URL: <https://doi.org/10.1109/icce-berlin50680.2020.9352169> (date of access: 05.03.2026).
53. Enhanced parking perception by multi-task fisheye cross-view transformers / A. Musabini et al. *IET Conference Proceedings CP887*. Stevenage, UK, 2024. P. 31–38. URL: <https://doi.org/10.1049/icp.2024.3273>.
54. Padilla R., Netto S. L., da Silva E. A. B. A Survey on Performance Metrics for Object-Detection Algorithms. *2020 International Conference on Systems, Signals and*

Image Processing (IWSSIP), Niterói, Brazil, 1–3 July 2020. 2020. URL: <https://doi.org/10.1109/iwssip48289.2020.9145130> (date of access: 25.03.2026).

55. The Pascal Visual Object Classes (VOC) Challenge / M. Everingham et al. *International Journal of Computer Vision*. 2009. Vol. 88, no. 2. P. 303–338. URL: <https://doi.org/10.1007/s11263-009-0275-4> (date of access: 25.03.2026).

56. Deep Learning for Camera Calibration and Beyond: A Survey / K. Liao et al. *ArXiv*. 2023. Abs/2303.10559. URL: <https://arxiv.org/abs/2303.10559>.

57. Kocur V., Ftáčnik M. Detection of 3D Bounding Boxes of Vehicles Using Perspective Transformation for Accurate Speed Measurement. *Machine Vision and Applications*. 2020. URL: <https://doi.org/10.48550/arXiv.2003.13137>.

58. Intelligent Parking Guidance System Using Computer Vision, IoT, and Edge Computing / P. Rajesh et al. *IEEE Open Journal of Intelligent Transportation Systems*. 2025. P. 1. URL: <https://doi.org/10.1109/ojits.2025.3605318> (date of access: 25.03.2026).

59. Real-Time Parking Space Management System Based on a Low-Power Embedded Platform / K. Kim et al. *Sensors*. 2025. Vol. 25, no. 22. P. 7009. URL: <https://doi.org/10.3390/s25227009> (date of access: 26.03.2026).

60. Avsiievych V., Павлова О., Mykhalchuk I. Security framework for cyber-physical smart parking systems with automated license plate recognition. *Cybersecurity: Education, Science, Technique*. 2025. Vol. 1, no. 29. P. 536–547. URL: <https://doi.org/10.28925/2663-4023.2025.29.900> (date of access: 22.04.2026).

61. Neural network based image recognition method for smart parking / O. Pavlova et al. *Computer systems and information technologies*. 2021. No. 1. P. 49–55. URL: <https://doi.org/10.31891/csit-2021-3-7> (date of access: 26.04.2026).

62. Авсієвич В., Коваленко В. Аналіз інформаційних технологій для розумної парковки на основі штучних нейронних мереж. *Актуальні Проблеми Комп'ютерних Наук (АПКН-2021)*. Хмельницький, Україна, 2021. С. 12–14.

63. Novorushchenko T., Pavlova O., Avsiievych V. Method of Assessing the Impact of External Factors on Geopositioning System Operation Using Android GPS API. *2021 IEEE 16th International Conference on Computer Sciences and Information*

Technologies (CSIT), LVIV, Ukraine, 22–25 September 2021. 2021. URL: <https://doi.org/10.1109/csit52700.2021.9648811> (date of access: 26.04.2026).

64. Convolutional Neural Network for Parking Slots Detection / P. Radiuk et al. *CEUR Workshop Proceedings*. 2022. P. 284–293.

65. Avsiyevych V., Kovalenko V. Cyber-physical system for smart parking based on computer vision technology. *Black Sea Science 2022: Proceedings of the International Competition of Student Scientific Works* / ed. by B. Iegorov, M. Mardar. Odesa, 2022. P. 335–346.

66. Авсієвич В., Кузьмін А. Дослідження вразливостей системи розумної парковки та способи їх усунення. *Актуальні Проблеми Комп'ютерних Наук (АПКН-2022)*. 2022. С. 11–14.

67. Avsiyevych V., Kawonga R. Security of smart parking cybersphysical system. *Information Technology & Engineering - 2023*. Mykolayiv, Ukraine, 2023. P. 59–61.

68. Павлова О., Авсієвич В., Кузьмін А. Дослідження факторів впливу на безпеку мобільних застосунків на прикладі клієнтської частини кіберфізичної системи розумної парковки. *Стан, досягнення та перспективи інформаційних систем і технологій*. Одеса, 2023. С. 98–99.

69. Savvidis P., Papakostas G. A. Edge Computing for Computer Vision in IoT: Feasibility and Directions. *EAI Endorsed Transactions on Internet of Things*. 2025. Vol. 11. URL: <https://doi.org/10.4108/eetiot.9404> (date of access: 26.04.2026).

70. Zhang Y., Si J., Si B. Integrative Approach for High-Speed Road Surface Monitoring: A Convergence of Robotics, Edge Computing, and Advanced Object Detection. *Applied Sciences*. 2024. Vol. 14, no. 5. P. 1868. URL: <https://doi.org/10.3390/app14051868> (date of access: 26.04.2026).

71. A Comparative Analysis of Pruning, Quantization, and Compilation for LightGBM-Based Electricity Anomaly Detection on IoT Edge Devices / S. Hadi et al. *Engineering, Technology & Applied Science Research*. 2026. Vol. 16, no. 2. P. 32935–32941. URL: <https://doi.org/10.48084/etasr.16433> (date of access: 26.04.2026).

72. Li Y., Huang Y., Tao Q. Improving real-time object detection in Internet-of-Things smart city traffic with YOLOv8-DSAF method. *Scientific Reports*. 2024. Vol. 14, no. 1. URL: <https://doi.org/10.1038/s41598-024-68115-1> (date of access: 26.04.2026).
73. PVswin-YOLOv8s: UAV-Based Pedestrian and Vehicle Detection for Traffic Management in Smart Cities Using Improved YOLOv8 / N. U. A. Tahir et al. *Drones*. 2024. Vol. 8, no. 3. P. 84. URL: <https://doi.org/10.3390/drones8030084> (date of access: 26.04.2026).
74. Lei S., Yi H., Sarmiento J. S. Synchronous End-to-End Vehicle Pedestrian Detection Algorithm Based on Improved YOLOv8 in Complex Scenarios. *Sensors*. 2024. Vol. 24, no. 18. P. 6116. URL: <https://doi.org/10.3390/s24186116> (date of access: 26.04.2026).
75. Analysis of vehicle and pedestrian detection effects of improved YOLOv8 model in drone-assisted urban traffic monitoring system / H. Dou et al. *PLOS ONE*. 2025. Vol. 20, no. 3. P. e0314817. URL: <https://doi.org/10.1371/journal.pone.0314817> (date of access: 26.04.2026).
76. Design and Implementation of a Real-Time Visual Tracking System for UAVs Based on PSDK / R. Yang et al. *Sensors*. 2026. Vol. 26, no. 7. P. 2145. URL: <https://doi.org/10.3390/s26072145> (date of access: 26.04.2026).
77. Teja Y. D. Static object detection for video surveillance. *Multimedia Tools and Applications*. 2023. URL: <https://doi.org/10.1007/s11042-023-14696-4> (date of access: 27.04.2026).
78. Attention-guided spatio-temporal feature fusion for robust video surveillance anomaly detection / S. D. Nivethika et al. *Scientific Reports*. 2026. URL: <https://doi.org/10.1038/s41598-026-36130-z> (date of access: 27.04.2026).
79. Real-Time Parking Space Management System Based on a Low-Power Embedded Platform / K. Kim et al. *Sensors*. 2025. Vol. 25, no. 22. P. 7009. URL: <https://doi.org/10.3390/s25227009> (date of access: 27.04.2026).
80. Edge computing for detection of ship and ship port from remote sensing images using YOLO / V. Sanikommu et al. *Frontiers in Artificial Intelligence*. 2025. Vol. 8. URL: <https://doi.org/10.3389/frai.2025.1508664> (date of access: 27.04.2026).

81. Resource-Efficient Design and Implementation of Real-Time Parking Monitoring System with Edge Device / J. Kim et al. *Sensors*. 2025. Vol. 25, no. 7. P. 2181. URL: <https://doi.org/10.3390/s25072181> (date of access: 27.04.2026).
82. M A., Juliet S., Anitha J. YOLO NAS-Powered Intelligent Vehicle Tracking and Monitoring System for Enhanced Parking Management in Urban Environments. *2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 14–15 March 2024. 2024. P. 2415–2420. URL: <https://doi.org/10.1109/icaccs60874.2024.10716939> (date of access: 27.04.2026).
83. Liu M., Wu B., Chen M. LFRE-YOLO: Lightweight Edge Computing Algorithm for Detecting External-Damage Objects on Transmission Lines. *Information*. 2025. Vol. 16, no. 12. P. 1035. URL: <https://doi.org/10.3390/info16121035> (date of access: 27.04.2026).

ДОДАТОК А

Лістинг програмного забезпечення кіберфізичної система визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторового аналізу

```

env = load_env(Path(__file__).resolve().parent / ".env")
rtsp_url      = env.get("RTSP_URL", "")
model_path    = env.get("MODEL_PATH", "yolov8n.pt")
conf          = float(env.get("CONF_THRESHOLD", "0.15"))
method        = env.get("SPATIAL_METHOD", "iomin")
t_iomin       = float(env.get("T_IOMIN", "0.50"))
n_buf         = int(env.get("N_BUF", "5"))
t_occ         = float(env.get("T_OCC", "0.6"))
t_free        = float(env.get("T_FREE", "0.3"))
config_path   = env.get("PARKING_CONFIG", "parking_config.json")
output_dir    = env.get("OUTPUT_DIR", ".")
jpeg_quality  = int(env.get("JPEG_QUALITY", "75"))
upload        = env.get("UPLOAD_ENABLED", "0") == "1"

if not rtsp_url:
    log.error("RTSP_URL not set in .env")
    sys.exit(1)

out = Path(output_dir)
out.mkdir(parents=True, exist_ok=True)

spots = load_spots(config_path)
spot_ids = [s.id for s in spots]
model = YOLO(model_path)
tf = TemporalFilter(spot_ids, n_buf, t_occ, t_free)

roi_mask: np.ndarray | None = None
roi_poly_scaled: np.ndarray = ROI_POLYGON.astype(np.int32)

os.environ["OPENCV_FFMPEG_LOGLEVEL"] = "-8"

def grab_fresh_frame(url: str, warmup: int = 5) -> np.ndarray |
None:
    cap = cv2.VideoCapture(url, cv2.CAP_FFMPEG)
    if not cap.isOpened():
        return None
    frame = None
    for _ in range(warmup):
        ret, f = cap.read()
        if ret:
            frame = f

```

```

    cap.release()
    return frame

    log.info("Core started. Method=%s, conf=%.2f, T_iomin=%.2f,
N_buf=%d",
            method, conf, t_iomin, n_buf)

while True:
    loop_start = time.monotonic()

    frame = grab_fresh_frame(rtsp_url)
    if frame is None:
        log.warning("Failed to grab frame, retrying in 3s...")
        time.sleep(3)
        continue

    if roi_mask is None:
        h, w = frame.shape[:2]
        ref_w, ref_h = REF_RESOLUTION
        sx, sy = w / ref_w, h / ref_h
        if abs(sx - 1.0) > 1e-3 or abs(sy - 1.0) > 1e-3:
            log.info("Frame %dx%d differs from reference %dx%d -
rescaling ROI and spots (sx=%.3f, sy=%.3f)",
                    w, h, ref_w, ref_h, sx, sy)
            roi_poly_scaled = np.stack([
                ROI_POLYGON[:, 0] * sx,
                ROI_POLYGON[:, 1] * sy,
            ], axis=1).astype(np.int32)
            spots = [
                ParkingSpot(
                    id=s.id,
                    polygon=np.stack([s.polygon[:, 0] * sx,
s.polygon[:, 1] * sy], axis=1),
                )
                for s in spots
            ]
            roi_mask = build_global_mask(frame.shape[:2],
roi_poly_scaled.tolist())
            log.info("ROI mask built: %dx%d", w, h)

        masked = cv2.bitwise_and(frame, frame, mask=roi_mask)
        dets = run_yolo(model, masked, conf)

        raw: dict[int, bool] = {}
        for spot in spots:
            if method == "iomin":
                raw[spot.id] = decide_iomin(dets, spot, t_iomin)
            else:
                raw[spot.id] = decide_vbase(dets, spot)

        states = tf.update(raw)
        rho = tf.get_rho()

```

```

free_count = sum(1 for v in states.values() if not v)
total = len(states)

state_data = {
    "timestamp": time.strftime("%Y-%m-%d %H:%M:%S"),
    "total": total,
    "free": free_count,
    "occupied": total - free_count,
    "method": method,
    "spots": {
        str(sid): {
            "occupied": states[sid],
            "rho": round(rho.get(sid, 0.0), 3),
        }
        for sid in spot_ids
    },
}
atomic_write_json(state_data, out / "state.json")

vis = draw_frame(frame, spots, states, dets)
tmp = out / f"temp_{os.getpid()}.jpg"
cv2.imwrite(str(tmp), vis, [int(cv2.IMWRITE_JPEG_QUALITY),
jpeg_quality])
atomic_replace_with_retry(tmp, out / "current.jpg")

if upload:
    upload_to_remote([out / "state.json", out /
"current.jpg"], env)

elapsed = time.monotonic() - loop_start
sleep_time = max(0.0, 1.0 - elapsed)
time.sleep(sleep_time)

```

ДОДАТОК Б

(обов'язковий)

Публікація



КІБЕРБЕЗПЕКА: освіта, наука, техніка

№ 1 (29), 2025

CYBERSECURITY:
EDUCATION, SCIENCE, TECHNIQUE

ISSN 2663 - 4023

DOI 10.28925/2663-4023.2025.29.900

UDC 004.056.53:004.94

Volodymyr Avsiievych

Master's degree, Student of Information Technologies Faculty
Khmelnyskyi National University, Khmelnytskyi, Ukraine
ORCID ID: 0009-0004-2040-8756
avsiievychvr@gmail.com

Olga Pavlova

PhD, Associate Professor, Head of the Department of Computer Engineering and Informational Systems
Khmelnyskyi National University, Khmelnytskyi, Ukraine
ORCID ID: 0000-0001-7019-0354
pavlovao@khmnu.edu.ua

Ihor Mykhalchuk

Assistant Lecturer of Computer Engineering and Informational Systems Department
Khmelnyskyi National University, Khmelnytskyi, Ukraine
ORCID ID: 0009-0002-5162-5986
mykhalchukiv@khmnu.edu.ua

SECURITY FRAMEWORK FOR CYBER-PHYSICAL SMART PARKING SYSTEMS WITH AUTOMATED LICENSE PLATE RECOGNITION

Abstract. Smart parking systems with automated license plate recognition (ALPR) are getting more popular in cities, but they have serious cybersecurity problems. This study analyzes security threats in smart parking infrastructure and offers ways to reduce them via vulnerability assessments and improved security designs. We analyzed various attacks such as protocol exploits and data interception risks in cyber-physical parking systems. Our research studies RTSP camera communication vulnerabilities, REST API security problems, and cloud service integration risks in license number recognition systems using computer vision technologies. Our approach includes vulnerability testing, threat modeling with STRIDE framework, penetration testing, and security analysis. We studied problems of RTSP camera protocol, HTTP/HTTPS communications, Laravel REST API setup, and Google Cloud Vision API integration. Results show that smart parking systems may have data interception risks, unauthorized access, API security problems and system integrity threats, which need multi-layered security approaches. We designed a cyber-physical parking system prototype with improved security measures in all components. The prototype has good license plate recognition accuracy while applying security methods without major performance reduction. Important security factors include secure communication protocols, encrypted transmission of data, authentication frameworks, input validation, rate limiting, and logging systems. This research helps to understand cybersecurity facets in IoT-based parking systems and offers methods for secure automated vehicle recognition setup in smart cities. This work is truly relevant for Ukrainian smart city projects, showing methodology applicable by IT companies for critical infrastructure protection.

Keywords: cybersecurity; cyber-physical systems; smart parking; automated license plate recognition; computer vision; IoT security; threat modeling; vulnerability assessment.

INTRODUCTION

Nowadays, urban infrastructure depends heavily on cyber-physical systems (CPS) that connect physical processes with computational elements through networked communication systems. Smart parking systems represent a critical area where cybersecurity becomes important — considering how sensitive vehicle identification data is and potential system abuse [14], [16], [17].



When automated license plate recognition (ALPR) technology is integrated with parking management systems, new attack surfaces appear that require thorough security analysis.

This research becomes relevant for Ukrainian smart city development projects. Critical infrastructure and cybersecurity have become a national priority during wartime conditions. Ukrainian cities actively apply digital transformation projects, including smart parking solutions. This makes setting up strong security frameworks from early deployment stages essential. Protecting civilian infrastructure systems such as transportation and parking management contributes to overall national cybersecurity strength.

Current smart parking systems often value functionality more than security factors, creating vulnerabilities that malicious actors can exploit. These vulnerabilities enable unauthorized access to vehicle data, manipulation of parking information, or complete system operation disruption. The combination of IoT devices, computer vision systems, and cloud-based services in smart parking architecture requires a serious cybersecurity approach addressing security measures.

Problem statement. This research is mainly related to the cybersecurity of smart parking systems and how to make efficient and secure infrastructure of these systems.

Analysis of recent research and publications. Looking at recent cybersecurity work on autonomous and connected vehicles, there's clearly growing worry about smart city infrastructure vulnerabilities [4]–[6]. Smart parking systems face unique security challenges — researchers have noted this is because of their distributed nature and how they depend on multiple communication protocols [7]–[10]. The problem is that systems often focus on making things work better while overlooking critical protection aspects.

Take Radiuk et al. [1] — they showed how well convolutional neural networks work for parking slot detection but completely missed data processing protection effects. It's the same story with automated license plate recognition studies [9], [10]. They focus mostly on accuracy improvements without thinking about cybersecurity — which is bad when you consider how sensitive license plate data is.

Earlier IoT protection framework research [31] gives us basic principles for securing networked infrastructures. But specialized approaches for parking facilities? Still not much there. Computer vision API integration [15] in smart parking brings additional concerns that need dedicated analysis [26]–[28]. This gap between making things work and making them secure keeps showing up in the literature.

Article aims. What we want to do in this research is develop and analyze cybersecurity measures for cyber-physical smart parking systems with automated license plate recognition. We are focusing on threat modeling, vulnerability assessment, and secure architecture design. Basically, we want to bridge that gap between making things work and making them secure in smart parking infrastructure.

THEORETICAL FOUNDATIONS OF RESEARCH

Cyber-physical systems in smart parking infrastructure bring together physical sensors, cameras, and actuators with computational processing and network communication capabilities [13]. When you're designing secure CPS, the theoretical foundations rely on defense-in-depth principles — multiple security layers that provide complete protection against different attack vectors. This makes sense because no single security measure will be enough in complex, interconnected systems.



Security assessment frameworks for CPS environments usually use risk analysis methods. They consider system components, potential vulnerabilities, and threat scenarios [29], [30]. A systematic approach means identifying critical system components $S = \{s_1, s_2, \dots, s_n\}$ and potential vulnerabilities $V = \{v_1, v_2, \dots, v_m\}$. Where components and vulnerabilities intersect, you get a risk matrix that guides security setup priorities.

You can formulate risk assessment mathematically:

$$R = \sum_{i=1}^n \sum_{j=1}^m P(v_j|s_i) \times I(v_j) \times L(s_i) \quad (1)$$

R represents overall system risk, $P(v_j|s_i)$ is the probability of vulnerability v_j being exploited given system component s_i , $I(v_j)$ represents impact severity of vulnerability v_j , and $L(s_i)$ indicates likelihood of component s_i being targeted.

Security effectiveness evaluation happens through complete testing and analysis of applied countermeasures. You can measure effectiveness like this:

$$E = 1 - \prod_{i=1}^k (1 - E_i) \quad (2)$$

Where E stands for overall security effectiveness and E_i is effectiveness of individual security control i across k total controls.

RESEARCH METHODOLOGY

Our research approach includes assessing vulnerabilities, modeling threats, and designing protection architecture for smart parking infrastructures. The experimental setup uses a new cyber-physical parking facility with multiple components that need evaluation.

Firstly, we need to do a network infrastructure analysis. We examine RTSP camera protocols, HTTP/HTTPS communications, and APIs. This analysis will show weak points in the communication system that could allow unauthorized access or data interception.

Threat modeling is the second component. We use the STRIDE method to find potential attack vectors. This means that we examine Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege threats in all components.

Thirdly, we do vulnerability testing. Penetration testing of interfaces includes camera access points, server APIs, and client applications [28]. This testing finds problems that a theoretical analysis might miss.

Finally, we'll determine the method of calculating metrics. We can use mathematical models to measure how well protection improvements work.

The experimental environment has a Hikvision DS-2CD1021-I camera [18], a Laravel REST API server [19], [21], and React Native mobile client [22], [23]. Each part of the system was tested on its own and as part of the whole to find problems that were either separate or affected the entire system.

Research was done at Khmelnytskyi National University using commonly available technologies and cloud services.

Fig. 1 shows the system architecture — how hardware components (CCTV camera), software components (Laravel REST API, Python OpenCV API), and client applications (React Native mobile app, web browser interface) interact. Each interaction point can be a potential attack surface that needs security consideration.



Smart Parking with Number Recognition Cyber-Physical System

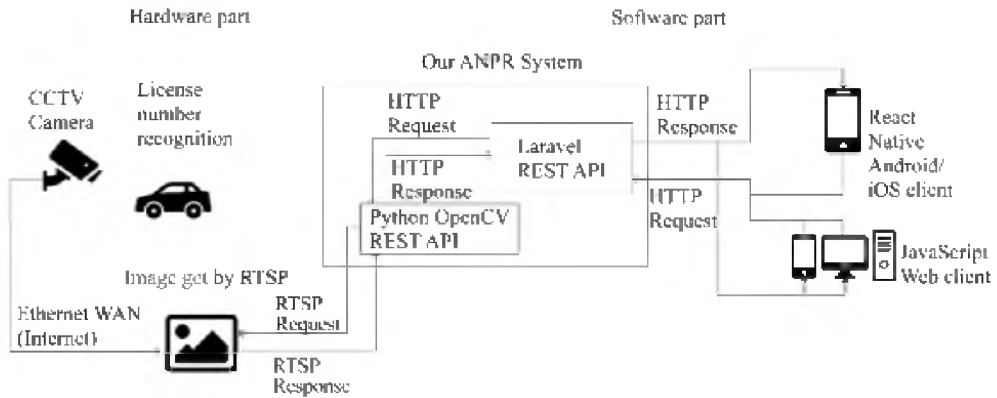


Fig. 1. Cyber-physical smart parking system architecture with ALPR capabilities

RESEARCH RESULTS

Security assessment of smart parking components

When we tested our cyber-physical smart parking system, we found several security factors in different system layers. Each layer of the system has unique vulnerabilities that need right security measures.

Looking at component interaction diagram (Fig. 2), we can see multiple communication endpoints that need security evaluation. HTTP connections between web/mobile clients and server hosting environments are especially important. The complexity of multi-component interaction in our system increases potential attack surfaces exponentially.

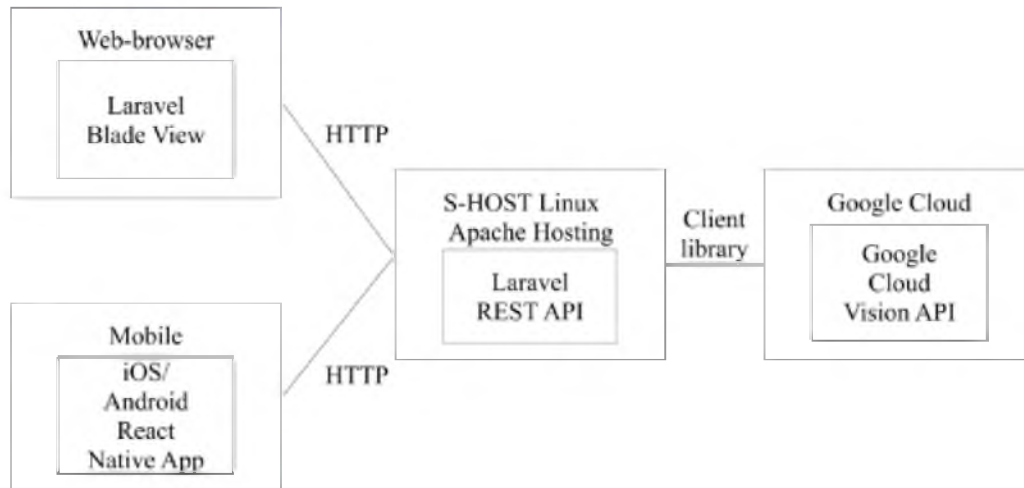


Fig. 2. Component interaction diagram



Camera network layer security

Usually, we use RTSP (Real-Time Streaming Protocol) for accessing camera feed. RTSP communications are significantly risky because of protocol characteristics. When we analyzed camera authentication methods, we found that default credentials and unencrypted data transmission create exploitable vulnerabilities. Many smart parking setups just don't bother changing default camera credentials — this creates easily exploitable entry points.

RTSP using devices must have strong authentication and encrypted data transfer to prevent unrestricted camera feed access. We can calculate the probability of unauthorized RTSP access like this:

$$P(UA) = f(P(DC), P(NA), P(PE)) \quad (3)$$

Where $P(DC)$ stands for probability of default credentials being unchanged, $P(NA)$ indicates network accessibility probability, and $P(PE)$ is protocol-level exploit probability. Relation f needs to be figured out experimentally. Studies show that IoT systems with cameras often keep default credentials, creating major vulnerabilities [11], [12].

API protection factors

We tested REST API setup using Laravel framework [19]–[21] for common web application issues. Our results show multiple vulnerability categories that need attention.

One of the most common measures that is used in these types of systems is token-based authentication. These tokens with strong encryption [25] provide secure session management while preventing unrestricted access. Developers need to carefully implement token generation, validation, and expiration policies.

We can't neglect input validation for image data. Processing this data prevents injection attacks and abuse. Validation includes file type verification, size limits, content analysis, and malware scanning. Sure, we know that data should be coming only from our local requests, but this can't be ignored if there is even the smallest possibility of something going wrong.

Rate limiting factors for external API calls of Google Cloud Vision API [15], [24] prevent denial-of-service attacks and excessive resource consumption.

These measures together provide serious protection against most common web application threats. We can determine combined effectiveness using standard evaluation method:

$$CE = 1 - \prod_{i=1}^n (1 - IE_i) \quad (4)$$

where IE_i is individual effectiveness of measure i .

Following established guidelines like OWASP recommendations [31] can greatly improve API protection posture. But generic recommendations need adaptation to specific smart parking needs.

Data processing protection

Google Cloud Vision API integration brings factors related to external service dependencies. Data processing protection means careful attention to data transmission, storage, and processing practices to minimize potential exposure risks. License plate data sensitivity needs particular attention to privacy and protection factors.

The image processing workflow (Fig. 3) shows data flow from clients through all system parts, including external API processing. Each stage has potential security problems that need specific protection measures. Data exposure risk increases at each processing stage, especially during external API communication.

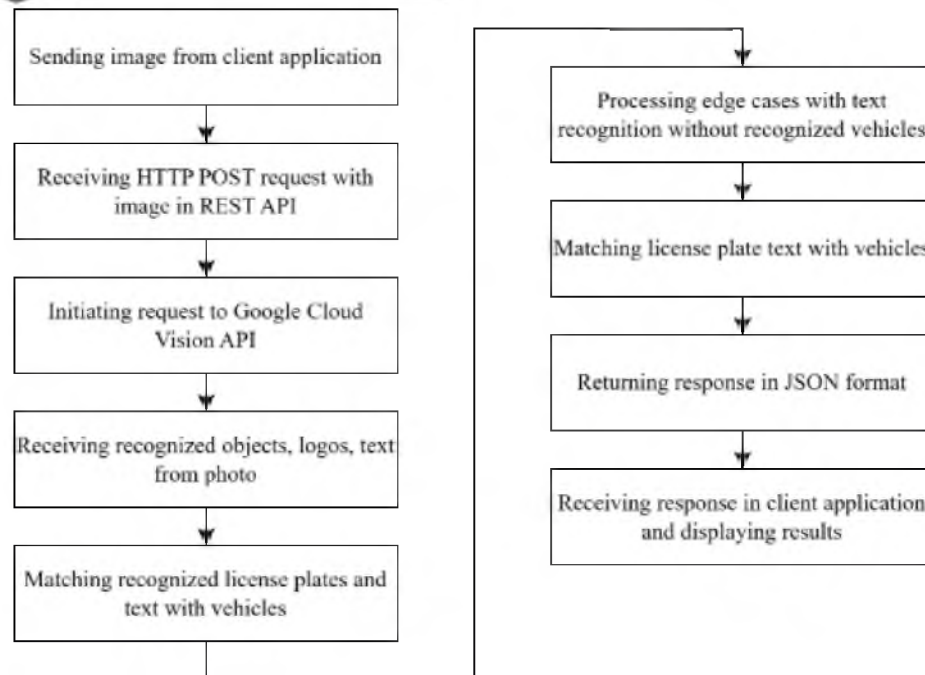


Fig. 3. Image processing workflow for automated license plate recognition system

Secure data transfer protocols and encryption standards are the main tools for protecting sensitive data during processing and transmission. We can determine data protection effectiveness through risk evaluation methods that consider exposure factors and processing complexity:

$$PrE = f(DER, PS, ES) \quad (5)$$

which is a formal signature description of relation between security context of system and its protection. Effective protection depends on minimizing data exposure risks (DER) across multiple processing stages (PS) while maintaining strong encryption standards (ES).

Network protection factors

Configuring a secure environment for accessing camera can greatly reduce RTSP vulnerability exposure. Virtual private networking and encrypted tunneling are perfect approaches for this task. Network segmentation (like DMZ) isolates camera traffic from other components, limiting potential attack spread.

We can calculate the network security factor for network-level protections:

$$NSF = 1 - (P(NI) \times P(PE)) \quad (6)$$

where P(NI) stands for probability of successful network interception and P(PE) indicates protocol-level abuse probability. Research shows that encrypted VPN tunnels and secured RTSP setups greatly reduce these vulnerability probabilities [29].

Data protection strategies

Data transfer protocols should encrypt license number (and other sensitive data) using algorithms like AES with appropriate key lengths. This measure includes cryptographically secure key generation, data integrity verification methods, and following proper key management practices.



Specifically, AES-256 encryption with responsibly managed keys provides strong data protection. Key rotation policies are needed to ensure security even if individual keys get compromised. We can evaluate data protection strength like this:

$$DPS = f(ES, KMQ) \quad (7)$$

where ES is strength of encryption algorithm and KMQ is a quality of key management. World-wide standards also recommend AES-256 encryption for sensitive data protection.

Performance factors

Protection enhancements usually affect performance — you need careful balance between defense and functionality. Performance factors must be checked in multiple areas:

Processing latency goes up because of encryption overhead. Protection setups generally introduce additional processing delays that must be considered in design.

License number recognition accuracy must stay at prominent levels despite protective measures. Computer vision should not experience performance variations when protection protocols are applied — the only one concern is decrypting image data.

Resource usage increases because of more processing needs for protective functions. It must be accounted for protection overhead in CPU and memory consumption.

Experimental setup observations

Our secure cyber-physical parking prototype shows several key capabilities. License number recognition maintains acceptable accuracy with active protection protocols. Results suggest that protective measures can successfully be integrated while preserving functional performance within acceptable ranges.

Testing shows that concurrent protective measure operation is achievable with proper design. System monitoring applies complete logging and capabilities that provide status information and threat detection.

The system achieves reliable productivity with protective measures enabled. Availability stays high during protection-enhanced operation.

Experience provides insights into practical factors for balancing protection needs with performance and functionality. The prototype shows that complete protection setup can be implemented in smart parking infrastructures when professionally designed and applied.

CONCLUSIONS AND FUTURE RESEARCH PROSPECTS

This research provides complete analysis of cybersecurity factors in cyber-physical smart parking systems with automated license plate recognition capabilities. Our study shows that smart parking setups face various security challenges requiring multifaceted security approaches.

The proposed security-conscious framework addresses vulnerabilities across network, application, and data protection layers while maintaining system functionality. Our setup proves that security measures can integrate into smart parking systems with careful attention to performance factors.

The practical contribution includes a security-conscious system architecture that addresses common vulnerabilities in smart parking setups. Research experience provides insights into balancing security needs with operational functionality. Vulnerability assessment methods provide systematic approaches for security improvement in similar systems.

Possible prospects for further research include direct applications like Ukrainian IT companies implementing the security framework in smart parking solution, educational



institutions integrating methodology into cybersecurity curricula, as well as local authorities benefit from established security requirements for smart parkings.

Key areas requiring further investigation include advanced threat modeling for computer vision systems, distributed security architectures for smart city infrastructure. Additionally, we should consider developing cybersecurity frameworks specifically adapted for Ukrainian post-war infrastructure reconstruction as it is critical research direction already.

The established methodology is not limited only to parking systems, but it can be extended to other cyber-physical infrastructure components, supporting comprehensive cybersecurity strategy development for smart city initiatives and digital transformation projects.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Radiuk, P., Pavlova, O., El Bouhissi, H., Avsiievych, V., & Kovalenko, V. (2022). Convolutional neural network for parking slots detection. *CEUR Workshop Proceedings*, 3156, 284–293. <https://hdl.handle.net/11300/26607>
2. Durlík, I., Miller, T., Kostecka, E., Zwierczewicz, Z., & Łobodzińska, A. (2024). Cybersecurity in autonomous vehicles—Are we ready for the challenge? *Electronics*, 13(13), 2654. <https://doi.org/10.3390/electronics13132654>
3. Higgins, M., Jha, D. N., Blundell, D., & Wallom, D. (2025). Security by design issues in autonomous vehicles. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2501.04104>
4. Guirrou, H., Youssef, T., Mohamed, Z. E., & Amal, T. (2024). Cybersecurity in autonomous vehicles: A comprehensive review study of cyber-attacks and AI-based solutions. *International Journal of Engineering Trends and Technology*, 72(1), 101–116. <https://doi.org/10.14445/22315381/IJETT-V72I1P111>
5. Kim, K., Kim, J. S., Jeong, S., Park, J.-H., & Kim, H. K. (2021). Cybersecurity for autonomous vehicles: Review of attacks and defense. *Computers & Security*, 103, 102150. <https://doi.org/10.1016/j.cosec.2020.102150>
6. Ghazali, A. A., & Fadzil, L. M. (2025). Intelligent illuminated parking system from cybersecurity perspectives: A review. *International Journal of Electrical and Electronic Engineering (IJEEE)*, 12(1). <https://doi.org/10.14445/23488379/IJEEE-V12I1P119>
7. Smart Parking Ltd. (2024, March 29). Innovative parking transformation. Retrieved from <https://www.smartparking.com/nz>
8. ZKTeco. (2024, April 20). ZKTecoParking. Retrieved from <https://www.zkteco.com/en/VideoParkingGuidanceSystem/ZKTecoParking>
9. OpenALPR. (2024, April 20). Automatic license plate recognition. Retrieved from <https://www.openalpr.com/>
10. OpenALPR. (2024, April 20). OpenALPR library. Retrieved from <https://github.com/openalpr/openalpr>
11. CVE-2025-30112: Bypass device pairing of 70mai Dashcam IS. (2025, March 24). Retrieved from <https://github.com/gco-chen/70mai/blob/main/README.md#finding-1---cve-2025-30112-bypass-device-pairing-of-70mai-dashcam-is>
12. CVE-2025-5113: Remote code execution on DivioTec IP Camera. (2025, June 3). Retrieved from <https://www.onekey.com/resource/security-advisory-remote-code-execution-on-diviotec-ip-camera-cve-2025-5113>
13. Kovalenko, V. V. (2022). *Cyber-physical smart parking system based on computer vision technology* (Master's thesis). Khmelnytskyi National University.
14. Avsiievych, V., & Kuzmin, A. (2022). Research of smart parking system vulnerabilities and ways to eliminate them. *Aktualni problemy komputernykh nauk (APKN-2022)*, 11–14. Khmelnytskyi National University.
15. Google Cloud. (2023, December 2). Vision AI. Retrieved from <https://cloud.google.com/vision>
16. Avsiievych, V., & Kawonga, R. (2023). Security of smart parking cyber-physical system. *Information Technology & Engineering – 2023*, 59–61. Mykolayiv, Ukraine.
17. Pavlova, O. O., Avsiievych, V. R., & Kuzmin, A. A. (2023). Research of factors influencing mobile application security on the example of the client part of a cyber-physical smart parking system. In *Stan, dosyahnennya ta perspektyvy informatsijnykh system i tekhnolohij: materialy XXIII Vseukrayinskoyi*



- naukovo-tekhnichnoi konferentsiyi molodykh vchenykh, aspirantiv ta studentiv* (pp. 98–99). ONTU Publishing House.
18. Hikvision Ukraine. (2024, April 20). Hikvision cameras with PoE support and outdoor installation. Retrieved from <https://hikvision.co.ua/ua/kamerv-vidconablyudeniva/ip-kamery/?ocf=F76S3V430F60S3V373>
 19. Laravel. (2024, February 19). The PHP framework for web artisans. Retrieved from <https://laravel.com/>
 20. Laravel. (2024, February 19). Laravel framework GitHub. Retrieved from <https://github.com/laravel/framework>
 21. TutorialsPoint. (2024, February 19). Laravel – overview. Retrieved from https://www.tutorialspoint.com/laravel/laravel_overview.htm
 22. Netguru. (2024, April 20). What is React Native. Retrieved from <https://www.netguru.com/glossary/react-native>
 23. Facebook. (2024, April 20). React Native. Retrieved from <https://github.com/facebook/react-native/blob/main/packages/react-native/scripts/cocoapods/helpers.rb>
 24. Google Cloud. (2024, April 20). Vision API. Retrieved from <https://cloud.google.com/vision?demo>
 25. Internet Engineering Task Force. (2024, December 15). RFC 7519: JSON Web Token (JWT). <https://tools.ietf.org/html/rfc7519>
 26. Elfaki, A. O., Messoudi, W., Bushnag, A., Abuzneid, S., & Alhmiedat, T. (2023). A smart real-time parking control and monitoring system. *Sensors*, 23(24), 9741. <https://doi.org/10.3390/s23249741>
 27. Alpina, A., Nikhil, K., Kunal, M., Shritej, N., & Shreenu, P. (2024). Parking the future: A review of IoT based parking systems. *International Research Journal of Modernization in Engineering Technology and Science*, 6(11), 2513–2517. <https://doi.org/10.56726/IRJMETS63971>
 28. Mohamed, E., Heba, A., Mahmoud, S. E., Anca, D. J., & Marianne, A. A. (2023). Intrusion detection for electric vehicle charging systems (EVCS). *Algorithms*, 16(2), 75. <https://doi.org/10.3390/a16020075>
 29. National Institute of Standards and Technology (NIST). (2024, December 15). *Cybersecurity framework: NIST special publication 800-53 Rev. 5*. Retrieved from <https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final>
 30. ISO/IEC. (2024, December 15). *ISO/IEC 27005:2018 – Information security risk management*. Retrieved from <https://www.iso.org/standard/75281.html>
 31. OWASP. (2024, December 15). *Internet of Things (IoT) Top 10*. Retrieved from <https://owasp.org/www-project-internet-of-things/>

**Авсієвич Володимир Русланович**

Студент магістратури факультету інформаційних технологій
Хмельницький національний університет, Хмельницький
ORCID ID: 0009-0004-2040-8756
avstievychvr@gmail.com

Павлова Ольга Олександрівна

Доктор філософії, доцент,
завідувач кафедри комп'ютерної інженерії та інформаційних систем
Хмельницький національний університет, Хмельницький
ORCID ID: 0000-0001-7019-0354
olya1607pavlova@gmail.com

Михальчук Ігор Володимирович

Викладач кафедри комп'ютерної інженерії та інформаційних систем
Хмельницький національний університет, Хмельницький
ORCID ID: 0009-0002-5162-5986
mykhailchukiv@khmnu.edu.ua

СИСТЕМА БЕЗПЕКИ ДЛЯ КІБЕР-ФІЗИЧНИХ РОЗУМНИХ СИСТЕМ ПАРКУВАННЯ З АВТОМАТИЧНИМ РОЗПІЗНАВАННЯМ НОМЕРНИХ ЗНАКІВ

Анотація. Розумні системи паркування з автоматичним розпізнаванням номерних знаків (ALPR) стають все більш популярними в містах, але вони мають серйозні проблеми з кібербезпекою. У цьому дослідженні проаналізовано загрози безпеці в інфраструктурі розумного паркування та запропоновано шляхи їх зменшення за допомогою оцінки вразливостей та вдосконалення дизайну безпеки. Ми проаналізували різні атаки, такі як вразливості протоколів та ризики перехоплення даних у кіберфізичних системах паркування. Ми вивчаємо вразливості зв'язку з камерами RTSP, проблеми безпеки REST API та ризики інтеграції хмарних сервісів у системах розпізнавання номерних знаків з використанням технологій комп'ютерного зору. Наш підхід включає тестування вразливостей, моделювання загроз за допомогою фреймворку STRIDE, тестування на проникнення та аналіз безпеки. Ми вивчили проблеми протоколу камер RTSP, HTTP/HTTPS-з'єднань, налаштування REST API Laravel та інтеграції API Google Cloud Vision. Результати показують, що розумні системи паркування можуть мати ризики перехоплення даних, несанкціонованого доступу, проблеми з безпекою API та загрози цілісності системи, які потребують багаторівневих підходів до безпеки. Ми розробили прототип кібер-фізичної системи паркування з покращеними заходами безпеки у всіх компонентах. Прототип має хорошу точність розпізнавання номерних знаків при застосуванні методів безпеки без значного зниження продуктивності. Важливими факторами безпеки є захищені протоколи зв'язку, зашифрована передача даних, системи автентифікації, валідації вхідних даних, обмеження швидкості та системи реєстрації. Це дослідження допомагає зрозуміти аспекти кібербезпеки в системах паркування на основі Інтернету речей і пропонує методи безпечного налаштування автоматичного розпізнавання транспортних засобів у розумних містах. Ця робота дійсно актуальна для українських проєктів розумних міст, оскільки демонструє методологію, яку застосовують ІТ-компанії для захисту критично важливої інфраструктури.

Ключові слова: кібербезпека; кіберфізичні системи; розумне паркування; автоматичне розпізнавання номерних знаків; комп'ютерний зір; безпека Інтернету речей; моделювання загроз; оцінка вразливості.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Radiuk, P., Pavlova, O., El Bouhissi, H., Avsiievych, V., & Kovalenko, V. (2022). Convolutional neural network for parking slots detection. *CEUR Workshop Proceedings*, 3156, 284–293. <https://hdl.handle.net/11300/26607>



2. Durlík, I., Miller, T., Kostecka, E., Zwierzewicz, Z., & Łobodzińska, A. (2024). Cybersecurity in autonomous vehicles—Are we ready for the challenge? *Electronics*, 13(13), 2654. <https://doi.org/10.3390/electronics13132654>
3. Higgins, M., Jha, D. N., Blundell, D., & Wallom, D. (2025). Security by design issues in autonomous vehicles. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2501.04104>
4. Guirrou, H., Youssef, T., Mohamed, Z. E., & Amal, T. (2024). Cybersecurity in autonomous vehicles: A comprehensive review study of cyber-attacks and AI-based solutions. *International Journal of Engineering Trends and Technology*, 72(1), 101–116. <https://doi.org/10.14445/22315381/IJETT-V72I1P111>
5. Kim, K., Kim, J. S., Jeong, S., Park, J.-H., & Kim, H. K. (2021). Cybersecurity for autonomous vehicles: Review of attacks and defense. *Computers & Security*, 103, 102150. <https://doi.org/10.1016/j.cosec.2020.102150>
6. Ghazali, A. A., & Fadzil, L. M. (2025). Intelligent illuminated parking system from cybersecurity perspectives: A review. *International Journal of Electrical and Electronic Engineering (IJEEE)*, 12(1). <https://doi.org/10.14445/23488379/IJEEE-V12I1P119>
7. Smart Parking Ltd. (2024, March 29). Innovative parking transformation. Retrieved from <https://www.smartparking.com/nz>
8. ZKTeco. (2024, April 20). ZKTecoParking. Retrieved from <https://www.zkteco.com/en/VideoParkingGuidanceSystem/ZKTecoParking>
9. OpenALPR. (2024, April 20). Automatic license plate recognition. Retrieved from <https://www.openalpr.com/>
10. OpenALPR. (2024, April 20). OpenALPR library. Retrieved from <https://github.com/openalpr/openalpr>
11. CVE-2025-30112: Bypass device pairing of 70mai Dashcam 1S. (2025, March 24). Retrieved from <https://github.com/gco-chen/70mai/blob/main/README.md#finding-1---cve-2025-30112-bypass-device-pairing-of-70mai-dashcam-1s>
12. CVE-2025-5113: Remote code execution on Diviotec IP Camera. (2025, June 3). Retrieved from <https://www.onekey.com/resource/security-advisory-remote-code-execution-on-diviotec-ip-camera-cve-2025-5113>
13. Kovalenko, V. V. (2022). *Cyber-physical smart parking system based on computer vision technology* (Master's thesis). Khmelnytskyi National University.
14. Avsiievych, V., & Kuzmin, A. (2022). Research of smart parking system vulnerabilities and ways to eliminate them. *Aktualni problemy komputernykh nauk (APKN-2022)*, 11–14. Khmelnytskyi National University.
15. Google Cloud. (2023, December 2). Vision AI. Retrieved from <https://cloud.google.com/vision>
16. Avsiievych, V., & Kawonga, R. (2023). Security of smart parking cyber-physical system. *Information Technology & Engineering – 2023*, 59–61. Mykolayiv, Ukraine.
17. Pavlova, O. O., Avsiievych, V. R., & Kuzmin, A. A. (2023). Research of factors influencing mobile application security on the example of the client part of a cyber-physical smart parking system. In *Stan, dosyahnennya ta perspektyvy informatsijnykh system i tekhnolohij: materialy XXIII Vseukrayinskoyi naukovo-tekhnichnoyi konferentsiyi molodykh vchenykh, aspirantiv ta studentiv* (pp. 98–99). ONTU Publishing House.
18. Hikvision Ukraine. (2024, April 20). Hikvision cameras with PoE support and outdoor installation. Retrieved from <https://hikvision.co.ua/ua/kamery-vidconablyudcniva/ip-kamery/?ocf=F76S3V430F60S3V373>
19. Laravel. (2024, February 19). The PHP framework for web artisans. Retrieved from <https://laravel.com/>
20. Laravel. (2024, February 19). Laravel framework GitHub. Retrieved from <https://github.com/laravel/framework>
21. TutorialsPoint. (2024, February 19). Laravel – overview. Retrieved from https://www.tutorialspoint.com/laravel/laravel_overview.htm
22. Netguru. (2024, April 20). What is React Native. Retrieved from <https://www.netguru.com/glossary/react-native>
23. Facebook. (2024, April 20). React Native. Retrieved from <https://github.com/facebook/react-native/blob/main/packages/react-native/scripts/cocoapods/helpers.rb>
24. Google Cloud. (2024, April 20). Vision API. Retrieved from <https://cloud.google.com/vision?demo>
25. Internet Engineering Task Force. (2024, December 15). RFC 7519: JSON Web Token (JWT). <https://tools.ietf.org/html/rfc7519>
26. Elfaki, A. O., Messoudi, W., Bushnag, A., Abuzneid, S., & Alhmiedat, T. (2023). A smart real-time parking control and monitoring system. *Sensors*, 23(24), 9741. <https://doi.org/10.3390/s23249741>



27. Alpana, A., Nikhil, K., Kunal, M., Shritej, N., & Shreenath, P. (2024). Parking the future: A review of IoT based parking systems. *International Research Journal of Modernization in Engineering Technology and Science*, 6(11), 2513–2517. <https://doi.org/10.56726/IRJMETS63971>
28. Mohamed, E., Heba, A., Mahmoud, S. E., Anca, D. J., & Marianne, A. A. (2023). Intrusion detection for electric vehicle charging systems (EVCS). *Algorithms*, 16(2), 75. <https://doi.org/10.3390/a16020075>
29. National Institute of Standards and Technology (NIST). (2024, December 15). *Cybersecurity framework: NIST special publication 800-53 Rev. 5*. Retrieved from <https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final>
30. ISO/IEC. (2024, December 15). *ISO/IEC 27005:2018 – Information security risk management*. Retrieved from <https://www.iso.org/standard/75281.html>
31. OWASP. (2024, December 15). *Internet of Things (IoT) Top 10*. Retrieved from <https://owasp.org/www-project-internet-of-things/>



This work is licensed under Creative Commons Attribution-noncommercial-sharelike 4.0 International License.

ДОДАТОК В
(обов'язковий)
Презентація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерної інженерії та інформаційних систем



**Кіберфізична система визначення
зайнятості паркувальних місць на
основі нейромережевого
розпізнавання об'єктів та
просторового аналізу**

Здобувач: Авсієвич Володимир Русланович
Науковий керівник: д.ф., доцент, зав. кафедри
Павлова Ольга Олександрівна

Хмельницький - 2026

МЕТА ДОСЛІДЖЕННЯ

Метою кваліфікаційної роботи магістра є вдосконалення існуючих методів визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторово-часового аналізу відеозображень для підвищення точності моніторингу та забезпечення роботи системи в умовах жорстко обмежених обчислювальних ресурсів.

Об'єктом дослідження є процес автоматизованого візуального моніторингу завантаженості паркувального майданчика.

Предметом дослідження є методи та алгоритми просторово-часового аналізу відеоданих на основі згорткових нейронних мереж.

ЗАДАЧІ ДОСЛІДЖЕННЯ

Поставлена мета досягається розв'язанням таких основних завдань:

- проаналізувати сучасний стан, існуючі апаратно-програмні комплекси та методи комп'ютерного зору у сфері візуального моніторингу паркувального простору;
- розробити математичні моделі просторового відбору об'єктів та часової стабілізації станів для компенсації ефекту візуального паралаксу;
- удосконалити метод просторово-часового аналізу відеоданих шляхом впровадження адаптивної метрики просторового перекриття та алгоритмів фільтрації на базі скінченного автомата з подвійним порогом;
- спроектувати та розробити програмне забезпечення кіберфізичної системи з урахуванням архітектурної вимоги щодо виконання нейромережевого прогнозування на центральному процесорі;
- провести експериментальні дослідження для оцінки точності розпізнавання та ресурсної ефективності оптимізованої системи в реальних умовах експлуатації.

НАУКОВА НОВИЗНА ТА ПРАКТИЧНА ЦІННІСТЬ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Наукова новизна отриманих результатів:

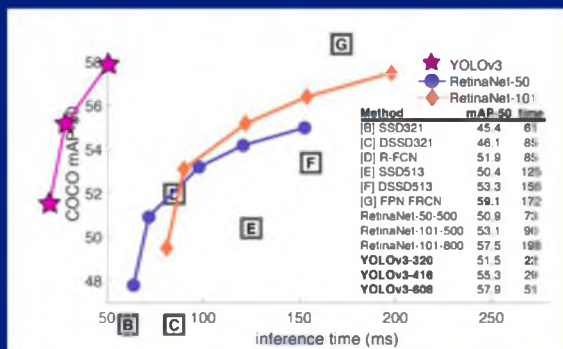
- набув подальшого розвитку метод просторово-часового аналізу для визначення зайнятості паркомісць, який використовує масштабно-інваріантну метрику просторового перекриття $IoMin$ у комбінації зі скінченим автоматом з подвійним порогом, що дозволило зменшити вплив перспективних спотворень та усунути коливання станів від маневруючого транспорту;
- удосконалено архітектуру програмного забезпечення кіберфізичної системи відеомоніторингу за рахунок впровадження моделі дискретного підключення до відеопотоку та алгоритмічних оптимізацій з забезпеченням відмовостійкості, що забезпечило її безперебійну роботу на обчислювальних потужностях центрального процесора без залучення дискретних графічних прискорювачів.

Практична цінність отриманих результатів полягає у розробленій апаратно-незалежній кіберфізичній системі візуального моніторингу паркувального простору.

АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

- ❑ Класичні системи розумних парковок використовують апаратні датчики, які є дороговартісними у розгортанні та обслуговуванні
- ❑ Системи на основі комп'ютерного зору вимагають наявності графічних прискорювачів для якісної роботи сучасних неймереж і мають проблему через паралакс і динамічні перешкоди
- ❑ При цьому на постах охорони парковок використовуються ПК з жорстко обмеженими ресурсами
- ❑ Таким чином, є актуальним розроблення методу просторово-часового аналізу на основі комп'ютерного зору, який компенсує вплив паралаксу і динамічних перешкод і здатний працювати виключно на потужностях центрального процесора.

АНАЛІЗ ВІДОМИХ МЕТОДІВ

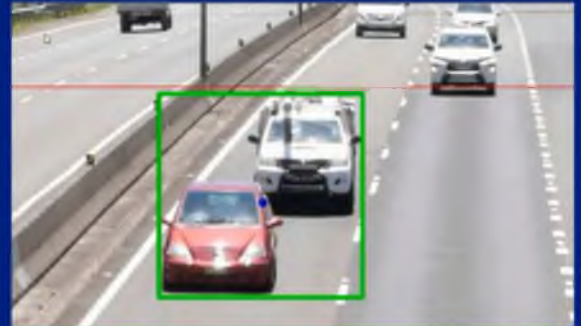


Графік залежності точності розпізнавання від швидкості обробки

- ❑ Традиційні методи (ГНГ, каскади Хаара) вразливі до перекриття об'єктів та змін освітлення
- ❑ Двостадійні ЗНМ (Faster R-CNN і т. д.) мають високу точність, але низьку швидкість (близько 5 FPS) і не підходять для роботи на CPU
- ❑ Одностадійні ЗНМ (YOLO, SSD) забезпечують оптимальний компроміс між швидкістю і точністю. Прогнозування за один прохід забезпечує роботу в реальному часі (до 80+ FPS)

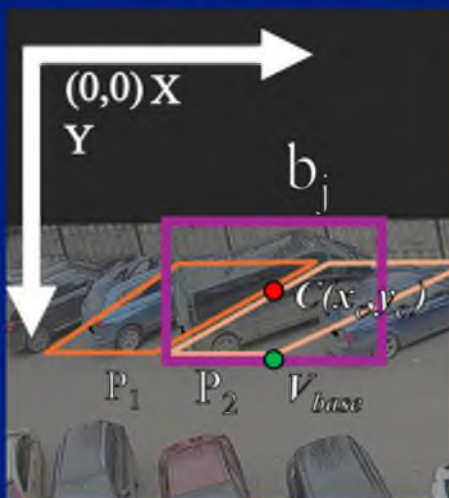
АНАЛІЗ ВІДОМИХ МЕТОДІВ

- ❑ Базова локалізація (рамки об'єктів) не дає інформації про розташування авто на паркомісці
- ❑ Метод центроїда легкий в реалізації, але критично вразливий до неточностей рамок та перспективних спотворень
- ❑ Базова метрика IoU надійніша, але має обмеження при суттєвій різниці габаритів автомобіля та площі паркомісця



Проблема перекриття

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ВИЗНАЧЕННЯ ЗАЙНЯТОСТІ ПАРКОМІСЦЬ



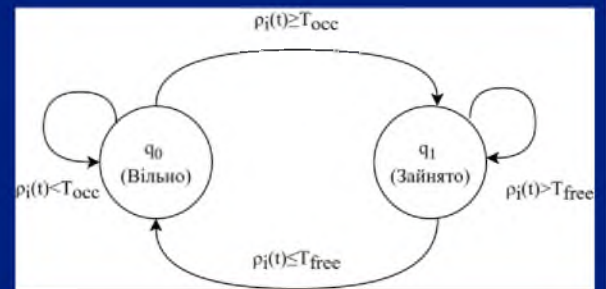
Компенсація ефекту паралаксу

- ❑ Перспективне спотворення зміщує геометричний центр рамки автомобіля на сусіднє паркомісце
- ❑ Рішення 1 - використання опорної точки нижньої грані $V_{base}(x_c, y_{max})$ як апроксимації контакту з дорогою
- ❑ Використання класичної метрика IoU дає похибки через різницю площ великих паркомісць і малих авто (або навпаки)
- ❑ Рішення 2 - масштабно-інваріантна метрику просторового перекриття IoMin

$$IoMin = \frac{S_{inter}}{\min(Area(b_j), Area(P_i))}$$

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ВИЗНАЧЕННЯ ЗАЙНЯТОСТІ ПАРКОМІСЦЬ

- Через тимчасове перекриття місць маневруючим транспортом або пішоходами виникає ефект коливання станів
- Рішення – впровадження моделі часової фільтрації на базі визначеного скінченного автомата з ковзним вікном
- Два пороги – підтвердження зайнятості (T_{free}) та звільнення (T_{occ})
- Поки немає перешкод створюється мертва зона, що гарантує відсутність хибних перемикань станів



Граф станів скінченного автомата системи



Абстрактно структурна модель потоку даних

МЕТОД ПРОСТОРОВО-ЧАСОВОГО АНАЛІЗУ ДЛЯ ВИЗНАЧЕННЯ ЗАЙНЯТОСТІ ПАРКОМІСЦЬ

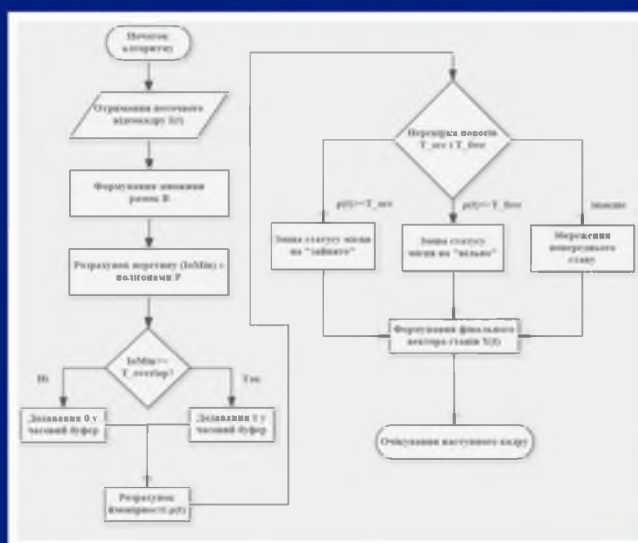


Схема методу



Маскування кадру

- Етапи алгоритму: маскування -> локалізація (YOLO) -> просторовий відбір (IoMin) -> часова стабілізація -> формування вектора станів
- Попередня обробка кадру – накладання попиксельної маски зони інтересу.
- Маскування знижує обчислювального навантаження на процесор та усуває хибні розпізнавань транспорту за межами майданчика

МЕТОД ПРОСТОРОВО-ЧАСОВОГО АНАЛІЗУ ДЛЯ ВИЗНАЧЕННЯ ЗАЙНЯТОСТІ ПАРКОМІСЦЬ

Метод	Precision	Recall	Accuracy
M1 (центр)	0,9598	0,8093	0,8698
V_{base}	0,9122	0,4260	0,6436
IoU_{slot} ($T=0,25$)	0,9021	0,9037	0,8874
$IoMin$ ($T=0,50$)	0,9147	0,8984	0,8926

- Обсяг тестування: 968 еталонних просторових станів за різних умов освітлення та заповненості
- Метод V_{base} має критичне мало значення повноти розпізнавання (42,6%) через недооцінювання паралаксу
- Метрика $IoMin$ успішно компенсує оптичні спотворення, досягає повноти 89,8% та точності 91,5%

Метрики роботи методів

Візуалізація роботи V_{base} (фіолетові точки)

МЕТОД ПРОСТОРОВО-ЧАСОВОГО АНАЛІЗУ ДЛЯ ВИЗНАЧЕННЯ ЗАЙНЯТОСТІ ПАРКОМІСЦЬ



Візуалізація роботи часового фільтру

Метод	Precision	Recall	Accuracy
M1 без фільтрації	0,9964	0,8900	0,9411
M1 + часовий фільтр	1,0000	0,8900	0,9428
IoU_{slot} без фільтрації	0,9058	0,9029	0,9007
IoU_{slot} + часовий фільтр	0,9103	0,8867	0,8956
$IoMin$ без фільтрації	0,9896	0,9256	0,9562
$IoMin$ + часовий фільтр	1,0000	0,9126	0,9545

Результати ефективності системи

- Через маневрування транспорту над вільними місцями виникають тимчасові хибні спрацювання
- Рішення: впровадження буфера у 5 кадрів і логіки подвійного порогу
- Таким чином, поглинаються короточасні перекриття і підвищується чіткість

ПРОГРАМНА РЕАЛІЗАЦІЯ

- Апаратно-незалежна кіберфізична система (Python, OpenCV, YOLOv8n, Flask)
- Поділ на ізольовані процеси – ядро і вебсервер
- Відмова від баз даних і використання атомарного заміщення файлів - цілісність даних та відсутність конфліктів

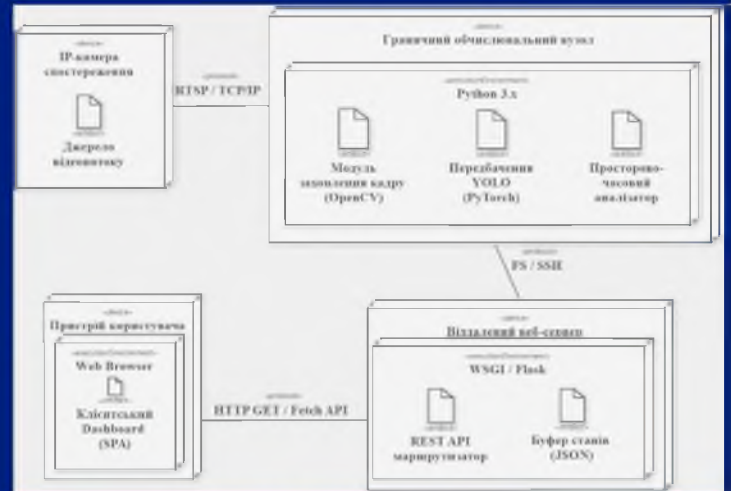
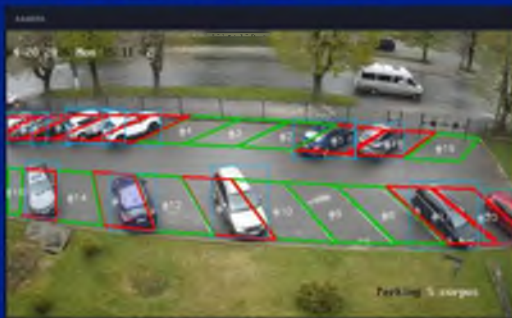


Схема архітектури системи

ПРОГРАМНА РЕАЛІЗАЦІЯ



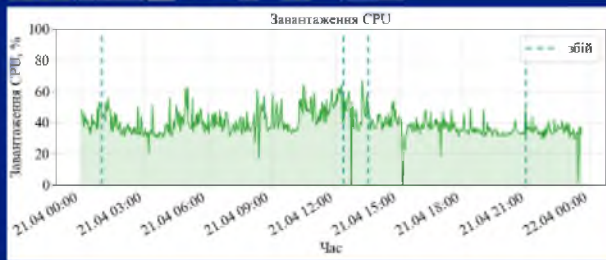
Анотований кадр



Статистична панель

- Інтерфейс користувача: інформаційна панель з індикацією станів
- Асинхронне оновлення даних без перезавантаження сторінки
- Відмовостійкість забезпечена постійним передпідключенням до камери у разі необхідності і перезапуском процесу системним планувальником

КС ПЕРИМЕНТАЛЬНА ОЦІНКА



Графік завантаження ЦП під час добового тесту

№сст	Режим потоку	Обмеження процесора	СРП (сер.), %	ОЗП (сер.), МБ	Час обробки кадру (сер.), мс
1	Основний 2560×1440	1 потік	8,12	414,7	22,30
2	Основний 2560×1440	Багатопотоковий	8,30	417,3	21,51
3	Вторинний 640×360	1 потік	6,12	354,1	22,35
4	Вторинний 640×360	Багатопотоковий	6,08	356,5	22,41

Споживання ресурсів на локальному стенді

- Середовище тестування: віртуалізований хостинг із квотами
- Оптимізації: перехід на субпотік камери, дискретне підключення (1 кадр/с), обмеження обчислень до 1 обчислювального потоку
- Максимальне споживання ОЗП близько 400 МБ, завантаження ЦП 40-60%

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень:

- проаналізовано сучасний стан, існуючі апаратно-програмні комплекси та методи комп'ютерного зору у сфері візуального моніторингу паркувального простору;
- розроблено математичні моделі просторового відбору об'єктів та часової стабілізації станів для компенсації ефекту візуального паралаксу;
- удосконалено метод просторово-часового аналізу відеоданих шляхом впровадження адаптивної метрики просторового перекриття та алгоритмів фільтрації на базі скінченного автомата з подвійним порогом;
- спроектовано та розроблено програмне забезпечення кіберфізичної системи з урахуванням архітектурної вимоги щодо виконання нейромережевого прогнозування на центральному процесорі;
- проведено експериментальні дослідження для оцінки точності розпізнавання та ресурсної ефективності оптимізованої системи в реальних умовах експлуатації.

Розроблений метод досягає показника повноти у 89,8% і чіткості у 91,47%, а з використанням часового фільтру досягає 100% чіткості. Розроблена кіберфізична система має середній показник споживання 40-60% ЦП і 400 МБ ОЗП.

ПУБЛІКАЦІЇ

- Авсієвич, В., Павлова, О., & Михальчук, І. (2025). СИСТЕМА БЕЗПЕКИ ДЛЯ КІБЕР-ФІЗИЧНИХ РОЗУМНИХ СИСТЕМ ПАРКУВАННЯ З АВТОМАТИЧНИМ РОЗПІЗНАВАННЯМ НОМЕРНИХ ЗНАКІВ. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 1(29), 536–547. <https://doi.org/10.28925/2663-4023.2025.29.900>

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Володимир АВСІЄВИЧ

Тема: Кіберфізична система визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторового аналізу

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень —; кількість сторінок записки 131

1. Короткий зміст роботи та прийнятих рішень У роботі розроблено та оптимізовано метод просторово-часового аналізу відеоданих для моніторингу паркувального простору. Спроектовано апаратно-незалежну кіберфізичну систему визначення зайнятості паркувальних місць.

2. Висновок про відповідність роботи дипломному завданню _____

Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз сучасних технологій моніторингу паркування та обґрунтовано вибір одностадійних згорткових нейронних мереж. У другому розділі розроблено математичні моделі просторового відбору об'єктів та часової стабілізації станів для компенсації ефекту візуального паралаксу. У третьому розділі удосконалено метод просторово-часового аналізу та проведено комп'ютерне моделювання. У четвертому розділі описано архітектуру та здійснено програмну реалізацію кіберфізичної системи, експериментально підтверджено її ефективність.

4. Позитивні сторони роботи: Яскраво виражена наукова новизна, практична цінність та адаптація сучасних алгоритмів комп'ютерного зору для граничних обчислень. Розроблена метрика IoMin ефективно компенсує перспективні спотворення при гострих кутах огляду камери. Розроблена архітектура кіберфізичної системи є надійною та гнучкою.

5. Негативні сторони роботи: В роботі недостатньо уваги приділено питанню масштабування системи для випадку одночасного моніторингу великого майданчика кількома IP-камерами з зонами перекриття видимості.

6. Оцінка графічного оформлення та пояснювальної записки роботи: —

7. Відгук про роботу в цілому: В загальному робота виконана на високому рівні.

8. Інші зауваження: —

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «відмінно» 100.00 (А)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) —

Борисюк Олександр Валодимирович, р.т.ч., проф. зав. каф. КІІ ХНУ

“ 1 травня ” 2026р.



Зав. кафедри КПС
д-р. філософії Ользі ПАВЛОВІЙ

Володимир АВСІЄВИЧ

ІІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2М-24-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений. Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений. Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Кіберфізична система визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторового аналізу
 Автор Володимир Авсієвич
 Освітня програма Комп'ютерна інженерія та програмування
 Рівень вищої освіти другий (магістерський)
 Спеціальність 123 Комп'ютерна інженерія
 Науковий керівник: доктор філософії, доцент Ольга ПАВЛОВА

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) 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	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах є збігом зі звітом з науково-дослідної практики автора Авсієвича Володимира «Аналіз сучасного стану технологій моніторингу паркувального простору», який було додано у репозиторій ХНУ 10 березня 2026 року;
 - 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
 - 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
 - 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
 - 5) значна частина знайденого плагіату відноситься до списку використаних джерел
- Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 3,98% і адресується до 46 першоджерел; та системою Anti-Plagiarism складає 21%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

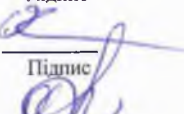
29.04.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


Підпис


Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Олег САВЕНКО
Ім'я, ПРІЗВИЩЕ

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Протокол аналізу звіту подібності експертом

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

Автор: Володимир АВСІЄВИЧ

Співавтор:

Назва: Кіберфізична система визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторового аналізу

Експерт: Ольга ПАВЛОВА

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 3.98%

Коефіцієнт подібності 2: 0.42%

Мікропробіли: 403

Заміна букв: 1

Інтервали: 0

Білі знаки: 6

Дата створення звіту: 2026-04-27 08:03:15.0

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:

2026-04-27

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 21.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилко в документах: 14%**

ID: 270692 Назва: МКР Кіберфізична система визначення зайнятості паркувальних місць на основі нейромережевого розпізнавання об'єктів та просторового аналізу Додано в БД: 2026-04-27 Автора: Володимир АВСІЄВИЧ Керівники: Ольга ПАВЛОВА Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	141135	1135	30990 (22%)	280 (25%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
269753	Назва: Звіт з ПДП АНАЛІЗ СУЧАСНОГО СТАНУ ТЕХНОЛОГІЙ МОНІТОРИНГУ ПАРКУВАЛЬНОГО ПРОСТОРУ Додано в БД: 2026-03-10 Автора: Авсієвича В.Р. Керівники: Павлова О.О. Консультанти: Опоненти:	29518 (21.0%)	263 (23.0%)