

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.032

До захисту допущено
Завідувач кафедри ММСА
О.Л. Тимошук
«__»_____ 2021 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз
на тему: «Нейронна мережа оптимальної роботи системи відеомоніторингу
паркувального простору»

Виконала:

студентка II курсу, групи КА-01мп
Білозорович Поліна Сергіївна

Керівник:

доцент кафедри ММСА
к.ф.-м.н., доц. Каніовська І.Ю.

Рецензент:

доцент кафедри математичного аналізу та теорії
ймовірностей КПІ ім. Ігоря Сікорського,
к.ф.-м.н., доц. Ільєнко М.К.

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка _____

Київ
2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
 «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
 ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
 ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
 КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти – другий (магістерський)
 Спеціальність – 124 «Системний аналіз»

ЗАТВЕРДЖУЮ
 Завідувач кафедри ММСА
 _____ О.Л. Тимошук
 «__» _____ 2021 р.

ЗАВДАННЯ

на магістерську дисертацію студентці Білозорович Поліні Сергіївні

1. Тема дисертації: «Нейронна мережа оптимальної роботи системи відеомоніторингу паркувального простору», науковий керівник дисертації Каніовська Ірина Юріївна, к.ф.-м.н., доцент, затверджені наказом по університету від 02 листопада 2021 р. № 3651-с

2. Термін подання студентом дисертації: 14 грудня 2021 р.

3. Об'єкт дослідження: методи пошуку та виявлення вільних місць на паркувальному просторі.

4. Предмет дослідження: моделі та архітектури нейронних мереж, та алгоритми розпізнавання об'єктів з відеопотоку.

5. Перелік завдань, які потрібно розробити:

1) дослідити сучасні системи виявлення вільних місць на паркувальному просторі;

2) провести огляд методів виявлення об'єктів в відеопотоку;

3) розробити систему на основі нової архітектури нейронної мережі для виявлення вільних місць на паркувальному просторі;

4) обрати дані для навчання та тестування мережі;

5) протестувати точність роботи розробленої моделі;

6) розробити стартап-проект;

7) зробити висновки за результатами наукового дослідження

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

- Ілюстрації до огляду існуючих систем та архітектур нейронних мереж;

- Зображення роботи системи та порівняння з існуючими аналогами.

7. Дата видачі завдання: 01 вересня 2021 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1	Вибір теми дисертації. Формулювання цілей та визначення завдань для виконання.	
2	Аналіз та порівняння існуючих систем та платформ для вирішення проблеми дослідження.	
3	Визначення підходу до вирішення проблеми та аналіз існуючих методів та алгоритмів.	
4	Створення архітектури та реалізація програмного продукту для вирішення визначеної проблеми.	
5	Стартап-проєкт.	
6	Висновки роботи програми та визначення подальшого розвитку продукту.	

Студентка

(підпис)

П.С. Білозорович

Науковий керівник дисертації

(підпис)

І.Ю. Каніовська

РЕФЕРАТ

Магістерська дисертація: 114 с., 23 рис., 11 табл., 1 дод., 14 джерел.

ОБРОБКА ВІДЕОПОТОКУ, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, НЕЙРОННА МЕРЕЖА, МАШИННЕ НАВЧАННЯ, АНАЛІЗ.

Тема роботи: нейронна мережа оптимальної роботи системи відеомоніторингу паркувального простору

Об'єкт дослідження: методи пошуку та виявлення вільних місць на паркувальному просторі.

Предмет дослідження: моделі й архітектури нейронних мереж та алгоритми розпізнавання об'єктів з відеопотоку. Використовується модифікований за допомогою архітектури метод нейронних мереж.

Метою даної магістерської дисертації є створення модифікованого за допомогою архітектури методу нейронних мереж, що дозволять швидко знаходити у відеопотоці паркувальні місця та визначати їх стан.

Актуальність даної теми зумовлена постійним збільшенням кількості машин на дорогах та невеликою кількістю виділених паркувальних місць. В роботі проведено огляд методів розпізнавання об'єктів з відеопотоку, розглянуто, існуючі методи знаходження вільних паркувальних місць в Україні та світі. Наведено аргументацію, чому обраний метод вирішення проблеми є оптимальним для України. Розглянуто та розв'язано декілька модельних задач, порівняно результати роботи алгоритмів.

Результати та їх новизна: виявлено застарілість та недосконалість методів знаходження вільних паркувальних місць в Україні. Запропоновано створення системи, що використовує нейронну мережу, яка дозволяє значно підвищити точність та швидкість виявлення вільного місця на паркувальному просторі.

ABSTRACT

Master's thesis: 114p., 23 figs., 11 tab., 1 appendix, 14 sources.

VIDEO STREAM PROCESSING, OBJECT RECOGNITION, THE NEURAL NETWORK, MACHINE LEARNING, ANALYSIS

The theme of the work is the neural network of optimal operation for video-based parking occupancy detection system.

Object of research: methods of finding and detecting free parking spots in the parking lot.

Subject of research: neural networks models and architectures and object recognition in a video stream algorithms. An architecture-modified neural network method is used.

The purpose of this master's thesis is to create an architecture-modified method of neural networks which will allow to find parking spaces via the video stream and determine the status.

The relevance of this topic is due to continuously increasing number of cars and the small number of parking spaces. The paper reviews the methods of recognizing objects from the video stream, considers the existing methods of finding free parking spaces in Ukraine and in the world. The thesis also argues why the chosen method of solving the problem is optimal for Ukraine. Several model problems are considered and solved, the results of algorithms are compared.

Results and their novelty: the obsolete and imperfect methods of finding free parking spaces in Ukraine are revealed. It is proposed to create a system that uses a neural network, which can significantly increase the accuracy and speed of detection of the free parking space.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	9
ВСТУП	10
РОЗДІЛ 1 ЗАДАЧА УПРАВЛІННЯ ТА МОНІТОРИНГУ ПАРКУВАЛЬНИХ МІСЦЬ. АНАЛІЗ ГОТОВИХ РІШЕНЬ.....	11
1.1 Вступ	11
1.2 Аналіз та порівняння платформ для організації паркувального простору	12
1.2.1 Паркувальна система від Parkeon.....	12
1.2.2 Паркувальна система Мікком AS101 ProPark.....	13
1.2.3 Система Smart parking	14
1.2.4 Паркувальна система від IEM.....	16
1.2.5 Паркувальні системи від CIRPARK.....	18
1.3 Аналіз та порівняння методів виявлення вільних місць на паркувальному просторі.....	21
1.3.1 Оптичне розпізнавання маркерів.....	21
1.3.2 Ультразвукові датчики	22
1.3.3 Петлевий детектор	23
1.3.4 Інфрачервоний паркувальний датчик	24
1.3.5 Магнітний датчик.....	26
1.4 Вибір методів для організації паркувального простору	27

1.5 Висновок	29
РОЗДІЛ 2 ЗАДАЧА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ З ВІДЕОПОТОКУ.	
АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ.....	30
2.1 Вступ.....	30
2.2 Огляд дескрипторів.....	30
2.2.1 Локальні дескриптори	30
2.2.2 Глобальні дескриптори.....	40
2.3 Розпізнавання вільних місць на паркувальній зоні з відеопотоку...	47
2.3.1 Аналіз основних компонентів.....	47
2.3.2 Навчання байєсівського класифікатора.....	47
2.3.3 Класифікації гістограм кольору	48
2.3.4 Навчання класифікатора SVM на основі текстур.....	50
2.3.5 CNN	52
2.4 Набори даних.....	60
2.4.1 PKLot	60
2.4.2 CNRPark + EXT	61
2.5 Висновки	63
РОЗДІЛ 3 ОПИС АРХІТЕКТУРИ ТА ПРОГРАМНОГО ПРОДУКТУ	
3.1 Визначення кордонів паркувальних місць	64
3.2 Визначення зайнятості паркувального місця.....	67
3.3 Запропонована архітектура	72
3.4 Результати роботи програми.....	74

3.5 Порівняння ефективності запропонованої архітектури з існуючою	75
3.6 Висновки	77
РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	79
4.1 Опис ідеї проєкту	79
4.2 Технологічний аудит ідеї проєкту.....	81
4.3 Аналіз ринкових можливостей запуску стартап-проєкту.....	82
4.4 Розроблення ринкової стратегії проєкту	93
4.5 Розроблення маркетингової програми стартап-проєкту	99
4.6 Висновки	103
ВИСНОВКИ.....	104
ПЕРЕЛІК ПОСИЛАНЬ	105
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	107

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

OCPP - Open Charge Point Protocol

DLM - Dynamic Load Management

HOG - Histogram of Oriented Gradients

SIFT - Scale-Invariant Feature Transform

SURF - Speeded Up Robust Features

CNN - Convolutional neural network

YOLO - You Only Live Once

HTTP - HyperText Transfer Protocol

LRN - Local Response Normalization

ВСТУП

Міське населення з кожним роком продовжує зростати, близько 54% населення планети в даний час проживає в містах, і до 2050 року ця цифра збільшиться до 66%, а мобільність громадян стає зростаючою проблемою з екологічними аспектами на кону. Міста шукають інтелектуальні рішення для того, щоб підвищити комфорт водіїв, які користуються індивідуальними транспортними засобами, зменшуючи при цьому його негативний вплив на навколишнє середовище. Для того, щоб якість життя не погіршувалася для його громадян, майбутнє місто має бути добре оснащене численними розумними рішеннями для безлічі проблем, що виникають у результаті масштабування.

У великих містах у автомобілістів часто виникає проблема парковки транспортного засобу. На те, щоб знайти вільне місце на звичайній парковці, може піти досить багато часу. Наприклад, в Німеччині пошук відповідного паркувального місця в середньому займає близько 10 хвилин. Згідно з дослідженням, яке проводилися компанією-лідером європейського ринку з управління парковками, водіям доводиться проїжджати до 4,5 км у пошуках місця, де можна припаркуватися. Дослідники з Каліфорнійського університету в Лос-Анджелесі повідомили, що в середньому 30% водіїв намагаються знайти місце для паркування в напружений час дня. Виходить, що водій втрачає не тільки час, а й гроші на паливо.

У роботі досліджуються деякі методики та пропонується вирішення проблеми виявлення місця для паркування. Це дослідження має на меті створити систему, яка вирішить проблему пошуку паркувального місця в Україні, використовуючи наявні ресурси.

РОЗДІЛ 1 ЗАДАЧА УПРАВЛІННЯ ТА МОНІТОРИНГУ ПАРКУВАЛЬНИХ МІСЦЬ. АНАЛІЗ ГОТОВИХ РІШЕНЬ

1.1 Вступ

У місті повинно бути достатньо місць для паркування, щоб забезпечити своїх мешканців та відвідувачів. Оскільки автомобілі є основним чинником транспорту, місто має задовольняти потреби водіїв. Якщо люди не можуть знайти паркувальне місце або якщо їм доведеться платити занадто багато, ймовірно, вони не повернуться вдруге. А мешканці, в свою чергу, повинні мати достатньо місця для паркування автомобіля поблизу свого будинку та робочого місця.

Рішення різних завдань в області організації дорожнього руху вимагає великих витрат ресурсів, тому одним з найважливіших підготовчих етапів є побудова моделі. В ході моделювання проводиться оцінка поточної ситуації, виявляються проблемні ділянки, виконується побудова різних варіантів рішення та оцінюється ефект, який буде досягнутий в результаті впровадження того чи іншого варіанту.

Для деяких задач вже існують платформи, які вирішують локальні проблеми.

1.2 Аналіз та порівняння платформ для організації паркувального простору

1.2.1 Паркувальна система від Parkeon

Parkeon є ключовим гравцем у сфері міської мобільності. Він працює у 60 країнах та прискорює зростання на ринках, що розвиваються у Східній Європі, Азії та Південній Америці. Це всесвітньо відомий інноваційний досвід у наданні паркувальної продукції, рішень, оперативних послуг та підтримки.

На сьогоднішній день у всьому світі поставлено та встановлено близько 185 000 паркувальних терміналів. Понад 95% терміналів Parkeon, що продаються сьогодні, працюють на сонячних батареях й повністю здатні підтримувати віддалені бездротові з'єднання з веб-системою управління Parkfolio, а також найновіші технології оплати та продажу квитків. [1]

Алгоритм роботи при в'їзді:

1. Машина зупиняється навпроти в'їзду, перед шлагбаумом.
2. Сенсор розпізнає присутність машин, а відповідна камера автоматично розпізнає і записує номер машини, дату і час в'їзду.
3. Дистанційна система управління отримує дані, надані цією камерою і додає їх в базу даних. Якщо клієнт вже зареєстрований, то додається також інформація про членство і дата резервування.
4. Шлагбаум на в'їзді піднімається, дозволяючи водієві потрапити на парковку. Шлагбаум автоматично опускається, коли машина проходить через сенсор безпеки.

Послідовність дій при виїзді з паркувального простору:

1. Машина зупиняється на виїзді, прямо перед шлагбаумом.
2. Сенсор розпізнає присутність машини, а камера перевіряє державний номер на відповідність інформації про платіж, або набору машин в базі даних.
3. Система управління перевіряє валідність коректного платежу, або дозволу на парковку без платежу, щоб дозволити виїзд.

4. Шлагбаум на виїзді піднімається, дозволяючи водієві покинути парковку.

5. Шлагбаум автоматично опускається, як тільки машина проїжджає через сенсор безпеки.

Така система є зручнішою за системи, у яких ключову роль відіграє людина. У такій реалізації також є перевага у фіксуванні та подальшому зберіганні номерів машин, адже цей функціонал може знадобитися як для збору бази відвідувачів у ТЦ, чи офісі, яку можна використовувати для маркетингових цілей, так і для полегшення пошуку машин у місті загалом. Проте задачу зі знаходження порожнього місця для паркування не вирішує.

З алгоритму зрозуміло, що дана реалізація використовує камеру та систему розпізнавання номерів, датчик для виявлення машини біля шлагбауму та базу даних для зберігання інформації про машини.

1.2.2 Паркувальна система Мікком AS101 ProPark

Мікком точно визначає місцезнаходження паркувальних місць та їх статус: вільне чи зайняте, також має функціонал підрахунку машин на парковці та машин, що переїхали кордон паркувальної зони. Для всіх місць впроваджено світлову індикацію та табло, які демонструють водію порожні місця й найкоротшу дорогу до цих місць. Система постійно та оперативно контролює стан паркувальної зони з наданням персоналу всієї необхідної інформації, управління табло, шлагбаумами та світлофорами.[2]

Система працює на базі комплексу безпеки AS101 Pro і є надійною, зручною і простою в експлуатації.

У проєкті створено нові пристрої (ультразвукові датчики, індикатори статусу місця, табло, датчики переїзду через лінію кордону паркувальної зони, суматори) і програмне забезпечення (рисунок 1.1).

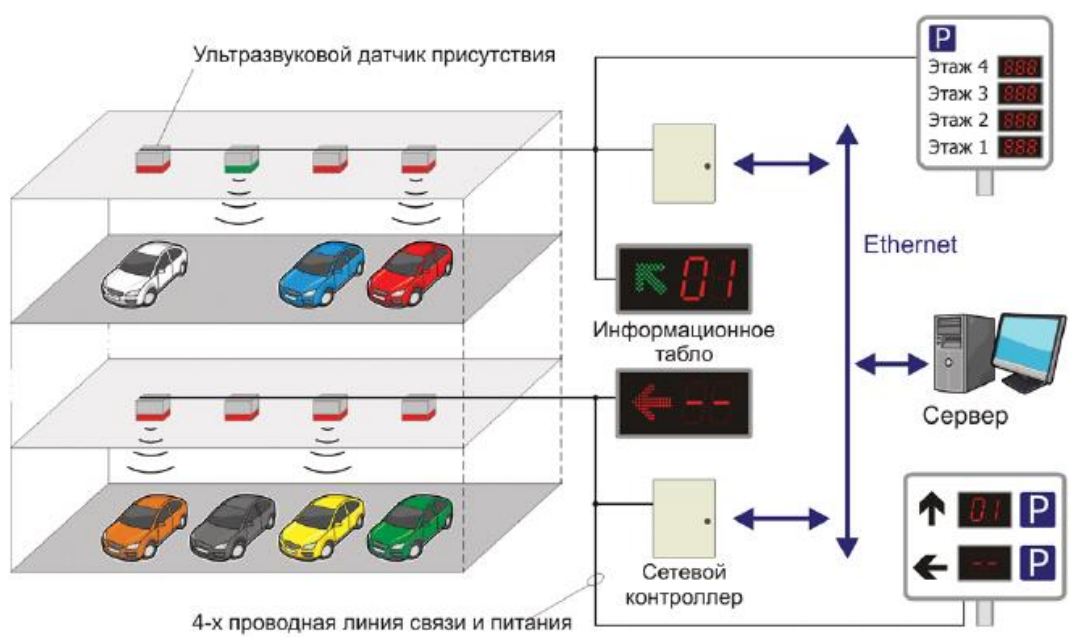


Рисунок 1.1– Схема роботи Мікком AS101 ProPark

Ця система є вже більш досконалою з точки зору підходу до вирішення задачі з пошуку вільного паркувального місця. Проте через велику кількість пристроїв, що застосовуються у даній реалізації, таке рішення підійде тільки для критих та обмежених паркувальних просторів. Отже поставлена задача вирішується тільки частково.

1.2.3 Система Smart parking

Система складається з ІТ-платформи і мережі стаціонарних камер, встановлених на опорах освітлення та будівлях (рисунок 1.2).

Smart parking автоматично зберігає всі припарковані поруч автомобілі. Інформація про транспортний засіб - реєстраційний номер, дата, час і місце паркування- автоматично надсилається до авторизованої інформаційної системи. [3]

Функції:

- 1) цілодобовий моніторинг в реальному часі;
- 2) визначення та сповіщення про порушення правил паркування і зупинки;
- 3) працювати з платними міськими парковками (зберігати час зупинки);
- 4) проводити статистичний аналіз паркувальної зони;
- 5) стежити за станом дорожніх знаків;
- 6) фіксувати машини за держномером, що внесені до бази, як правопорушники.

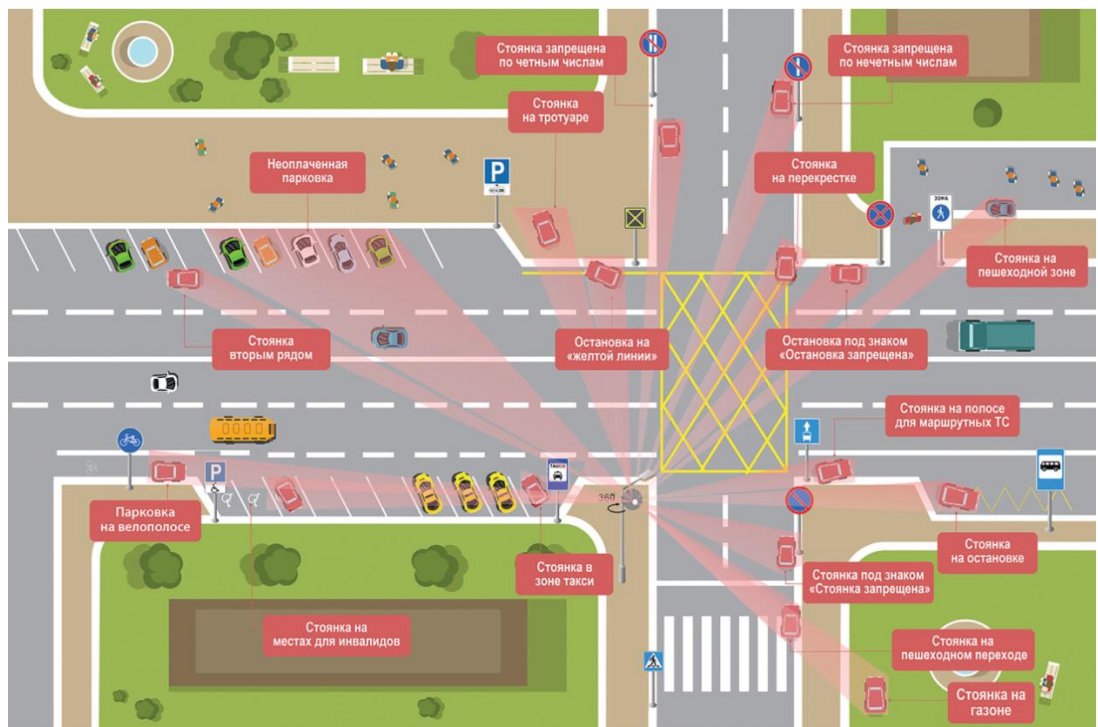


Рисунок 1.2– Схема роботи Smart parking

Smart parking – ще одне цікаве рішення. Система є доволі бюджетною, адже використовує тільки камери, які за допомогою нейронних мереж фіксують та передають багато корисної інформації. Але задача з виявлення вільних місць, нажаль, залишається невирішеною.

Ідея застосовувати камери для розпізнавання місцезнаходження автомобілів та стану паркувальних місць є цікавою з точки зору пошуку універсального підходу до пошуку вільного слоту для авто.

1.2.4 Паркувальна система від IEM

Завдяки впровадженню вуличних датчиків, IEM забезпечує точний огляд будь-якої вуличної парковки.

Датчики PrestoSense другого покоління мають подвійне виявлення, як магнітне, так і ультразвукове. Наявність або відсутність транспортного засобу визначається і передається в режимі реального часу з 2-секундним циклом виявлення. Надійність цієї запатентованої технології досягає 99,5%. [4]

Детектори PrestoSense використовують технологію Internet of Things на основі LoRaWANTM (широкополосна мережа з низькою потужністю) для спілкування з платформою для паркування. Вони виявляють не тільки присутність автомобіля, але і тривалість стоянки. Інформація передається оператору паркування. Він використовує дані для аналізу завантаженості та оборотності свого автопарку та для оптимізації роботи. Автомобіліст використовує інформацію безпосередньо на своєму смартфоні. Міська система керування дорожнім рухом або система GPS автомобіліста направляють його до наступної доступної стоянки.

Стан детектора передається на платформу стоянки IEM. Окрім даних про наявність, платформа збирає платіжні дані з парктрометрів та віртуальні платіжні дані з додатків для смартфонів. Ця інформація обробляється та аналізується на порталі операторів Presto1000.

Presto1000 надає огляд ситуації з паркуванням у містах. Складні звіти інформують оператора про зайнятість та поведінку автомобіліста.

Детектори PrestoSense надають дані про завантаженість міських вуличних паркувальних місць у режимі реального часу. Вони використовуються містом та користувачами різних додатків для кращого використання паркування.

У даній системі (рисунок 1.3) використано більш функціональні датчики. Та навідміну від тих, що пропонувались у системі Мікком AS101 ProPark, ці датчики можна використовувати не тільки на критих паркувальних просторах, а й просто вздовж дороги.

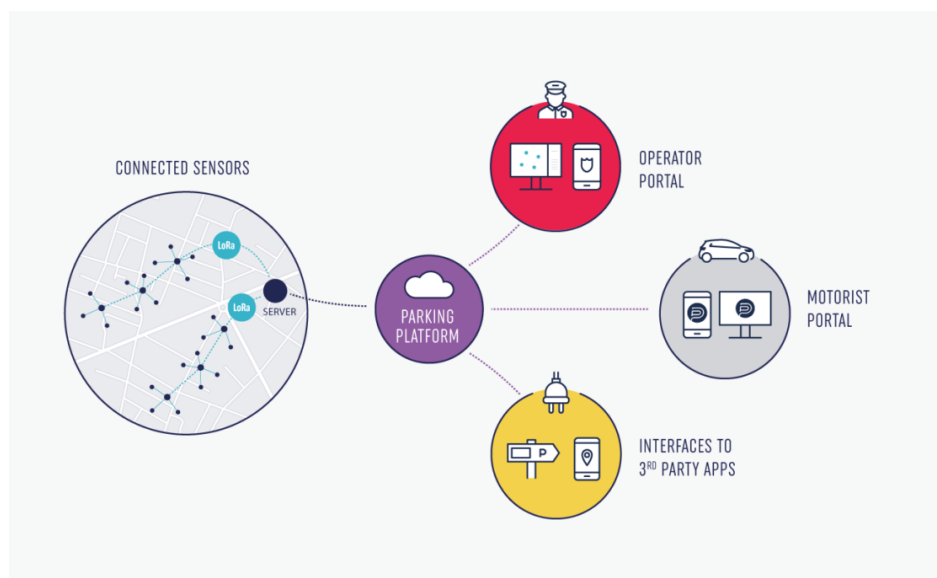


Рисунок 1.3– Схема роботи ІЕМ

Та все ще залишається актуальним один з головних недоліків подібних рішень – висока ціна для масштабного впровадження.

1.2.5 Паркувальні системи від CIRPARK

Це потужне рішення, яке інтегрує системи iPark, LEDPark та EVPark.

Платформа, виготовлена з програмного забезпечення CirPark Scada та сторонньої інтеграції. [5]

iPark-одна з найбільш вражаючих і довговічних систем на ринку систем навігації, пошуку автомобілів та підрахунку.

Система підрахунку рівня та площі з повним набором детекторів та інформацією на панелі для паркування в приміщеннях та на вулиці.

Система динамічного орієнтування в приміщенні/на відкритому повітрі, яка управляє інформацією користувача з метою оптимізації зайнятості та руху транспортних засобів (рисунок 1.4).

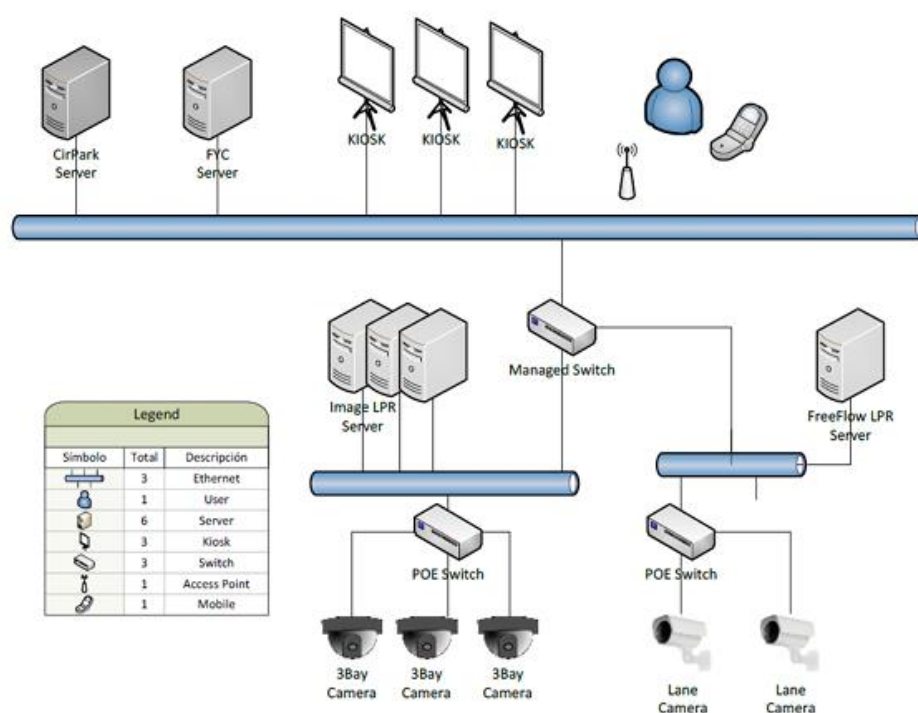


Рисунок 1.4– Схема роботи iPark

Потужна система, здатна запропонувати рішення для пошуку автомобілів на основі QR-коду або розпізнавання номерних знаків у межах смуг або на кожному паркувальному місці, пропонуючи користувачам розташування та маршрут до власного автомобіля за допомогою програми користувача.

Регульована система світлодіодного освітлення зі LED технологією, інтегрована з системою керування парковкою. Регульоване світлодіодне обладнання з низьким енергоспоживанням (рисунок 1.5). Вбудований у платформу CirPark для повного автоматичного управління.

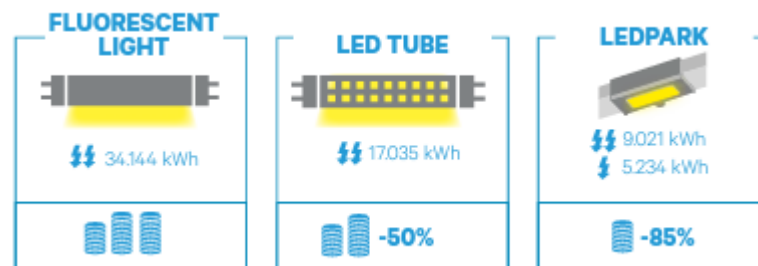


Рисунок 1.5– Схема роботи iPark

EVPark - це рішення для електромобілів.

EVPark пропонує широкий асортимент зарядних пристроїв для електромобілів; настінне/наземне кріплення, повільна/швидка зарядка та одинарна/подвійна розетка. Для приміщень/на відкритому повітрі.

Щоб забезпечити дружню експлуатацію зарядних пристроїв користувачами та вигідну бізнес-модель для оператора паркування, рішення EVPark використовують OCPP (Open Charge Point Protocol), широко поширений у сфері електромобільності.

Систему Dynamic Load Management (DLM) можна інтегрувати з платформою CirPark, пропонуючи найбільш повне рішення, наявне на ринку. Dynamic Load Management (DLM) - це програмне рішення, розроблене для управління енергією, коли одночасно працює кілька зарядних станцій. DLM

дозволяє заряджати декілька електромобілів одночасно найбільш ефективним способом, динамічно використовуючи залишкову потужність і балансуючи її між зарядними пристроями (рисунк 1.6). Це також дозволяє збільшити кількість зарядних станцій без збільшення контрактної потужності.

Таким чином, DLM можна встановити на місцях, де електроустановка повністю розподілена на електромобілі, або на місцях, де інший об'єкт ділить максимальну доступну потужність.

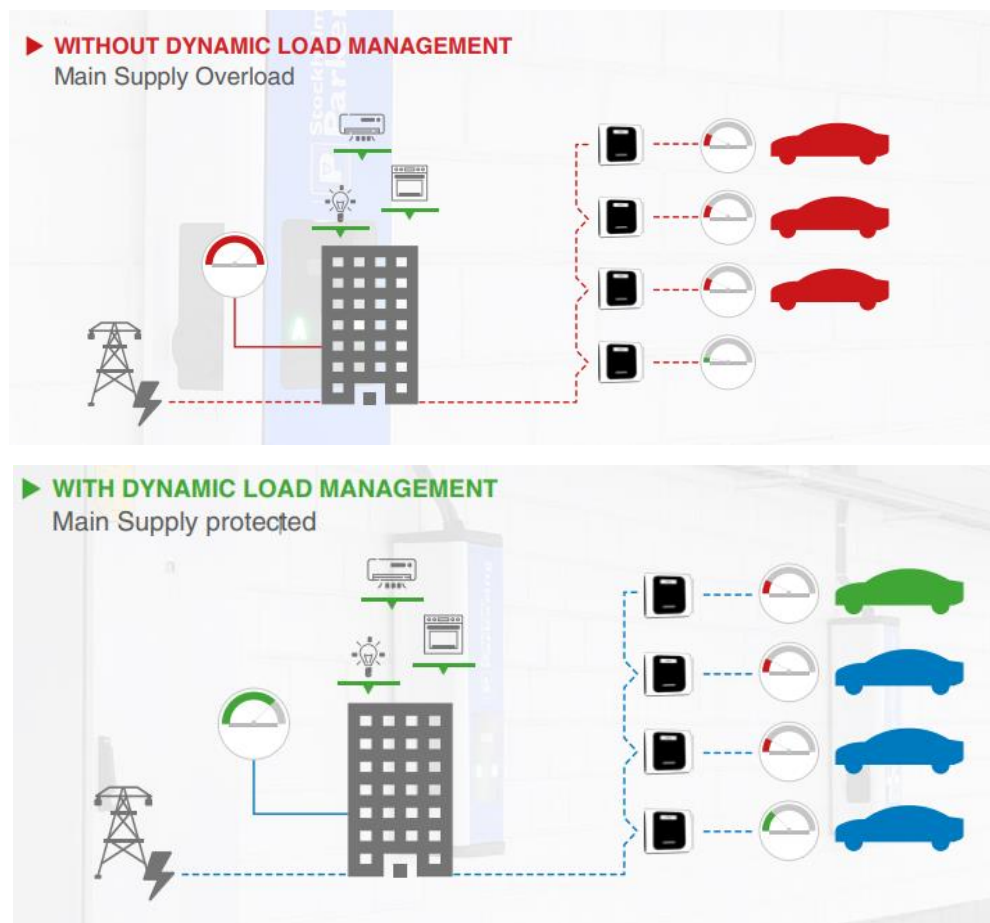


Рисунок 1.6– Різниця роботи системи без та з DLM

Компанія, що пропонує дане рішення, використовує гнучку схему реалізації. Спеціалісти підбирають зі свого арсеналу пристрої, що більш підійдуть під потреби замовника. Функціональність та масштабність системи вражає, але витрати на подібну систему знадобляться чималі.

1.3 Аналіз та порівняння методів виявлення вільних місць на паркувальному просторі

Отже, з оглянутих вище систем можна виділити наступні методи для визначення стану паркувальних місць:

- оптичне розпізнавання;
- ультразвукові датчики;
- петлевий детектор;
- інфрачервоний паркувальний датчик;
- магнітний датчик.

1.3.1 Оптичне розпізнавання маркерів

Кожне місце для паркування зазначено прямокутником-маркером. Якщо автомобіль займає місце для паркування, то колір всередині прямокутника зміниться і можна вважати це місце зайнятим. [6]

Для даного підходу необхідно, щоб всі паркувальні місця були під відеоспостереженням. Якщо камер на парковці досить, то технологія оптичного розпізнавання не зажадає установки додаткового обладнання. Даний підхід не підійде для відкритих парковок, так як сніг і калюжі можуть перекрити графічні позначки (рисунок 1.7).



Рисунок 1.7– Знімок екрану під час оптичне розпізнавання маркерів

1.3.2 Ультразвукові датчики

Хоча більшість водіїв покладаються на ці датчики для паралельного паркування, ультразвукові датчики використовують також для сканування, збору та передачі інформації про кількість паркувальних місць у режимі реального часу до хмари, що, врешті-решт, допомагає водіям знайти доступні місця для паркування.

Коли автомобіль рухається вулицею, ультразвукові датчики передають звукові хвилі та збирають дані про припарковані автомобілі та порожні місця (рисунок 1.8). Потім ці дані анонімно надсилаються до хмари для аналізу та об'єднання з нашим існуючим механізмом прогнозування доступності паркування для визначення рівнів заповнюваності блок за блоком. [7]

Для реалізації такої паркувальної системи кожне місце необхідно буде обладнати бездротовими датчиками, які будуть визначати вільно місце чи ні.

У датчиках необхідно періодично міняти батарейки, крім того їх необхідно встановити так щоб їх не могли пошкодити.

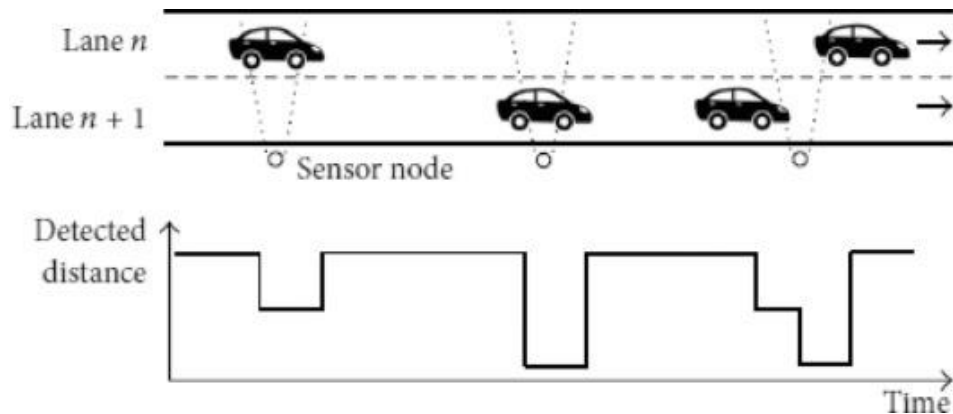


Рисунок 1.8– Принцип роботи ультразвукових датчиків

1.3.3 Петлевий детектор

Зона виявлення петлевого детектору максимальна, дешева і нечутлива до будь-яких кліматичних умов. Петльові детектори долають проблему монтажу датчика з опорними поперечними кронштейнами. Основні компоненти системи індуктивної петлі включають один або кілька витків ізолюваного петлевого дроту, намотаного в неглибоку щілину, випиляну на тротуарі біля в'їзду на автостоянку та на місцях для паркування. Щоразу, коли транспортний засіб переходить через петлю або зупиняється в межах петлі, індуктивність петлі зменшується. [7]

У тротуар встановлюється система замкнутого дроту, що заглиблена в землю, і прикріплюється до сповіщувача транспортного засобу. Детектор з живленням посилає невелику кількість змінного струму через цю петлю. Струм створює поле індуктивності в контурі.

Детектор шлейфа автомобіля визначає транспортний засіб і активує релейний вихід.

Елемент всередині детектора автомобіля вимірює частоту поля індуктивності. Коли транспортний засіб проходить через належним чином встановлене та активне поле, відбувається інтенсивне падіння виміряної індуктивності. Детектор відчуває падіння і спрацьовує на виході, як реле, до системи контролю доступу до транспортного засобу, наприклад, для управління воротами (рисунок 1.9).

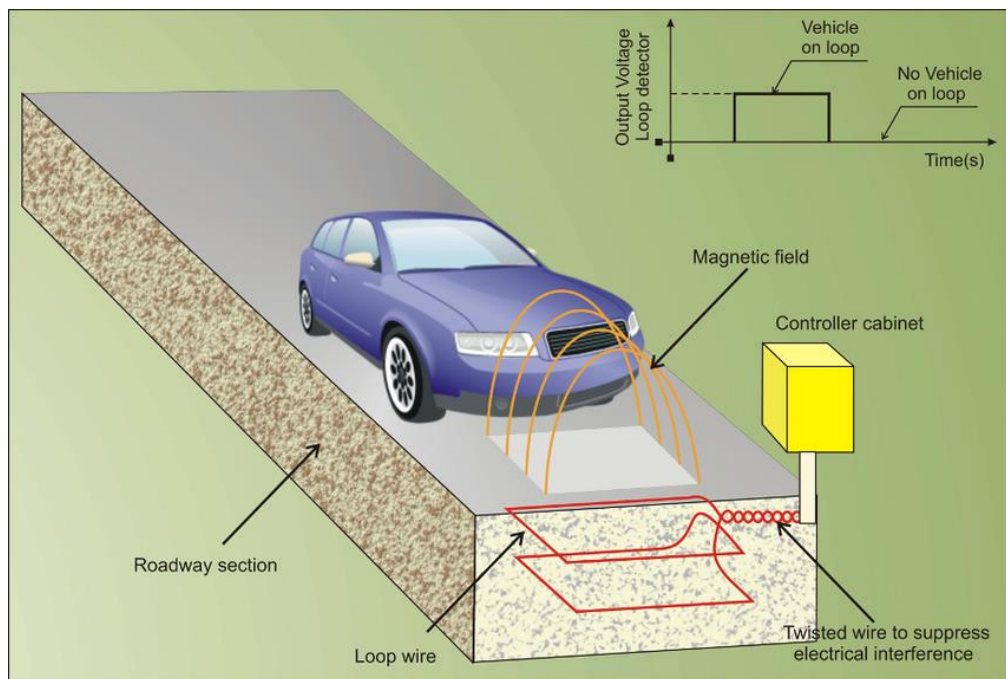


Рисунок 1.9– Принцип роботи петлевого детектору

1.3.4 Інфрачервоний паркувальний датчик

Коли автомобіль блокує інфрачервону смугу, приймач фіксує появу машини на паркувальному місці. Після того, як автомобіль звільниться від світла, приймач зафіксує паркувальне місце, як вільне.

Такий датчик забезпечує просте, недороге та надійне виявлення транспортних засобів та вказівки щодо паркування. Виключно спираючись на пряме світло між передавачем та індикатором, він забезпечує найвищу точність будь-якого датчика у своєму класі (рисунок 1.10). [8]

Однією з найважливіших переваг інфрачервоних датчиків у порівнянні з ультразвуковими датчиками є те, що інфрачервоне світло розсіюється від усіх об'єктів. Підлога служить еталонною областю для датчиків. Якщо, наприклад, ультразвуковий датчик не працює, якщо він потрапляє на похилі поверхні, такі як лобове скло автомобіля, інфрачервоний датчик вловлює відбиття навіть з похилого положення.

Ще одна особливість інфрачервоного датчика RTV полягає в тому, що місця для паркування, розташовані поруч, можуть мати два датчики, але лише один світлодіодний дисплей. Якщо одне місце для паркування зайняте, дисплей зеленого кольору, а якщо обидва місця зайняті, дисплей червоного кольору. Оскільки світлодіоди багатокольорні, кольори можна вибирати, виходячи з уподобань операторів паркування.

Використовуючи технологію Bluetooth, ця проста в установці система легко інтегрується з паркувальним менеджером для максимальної універсальності.

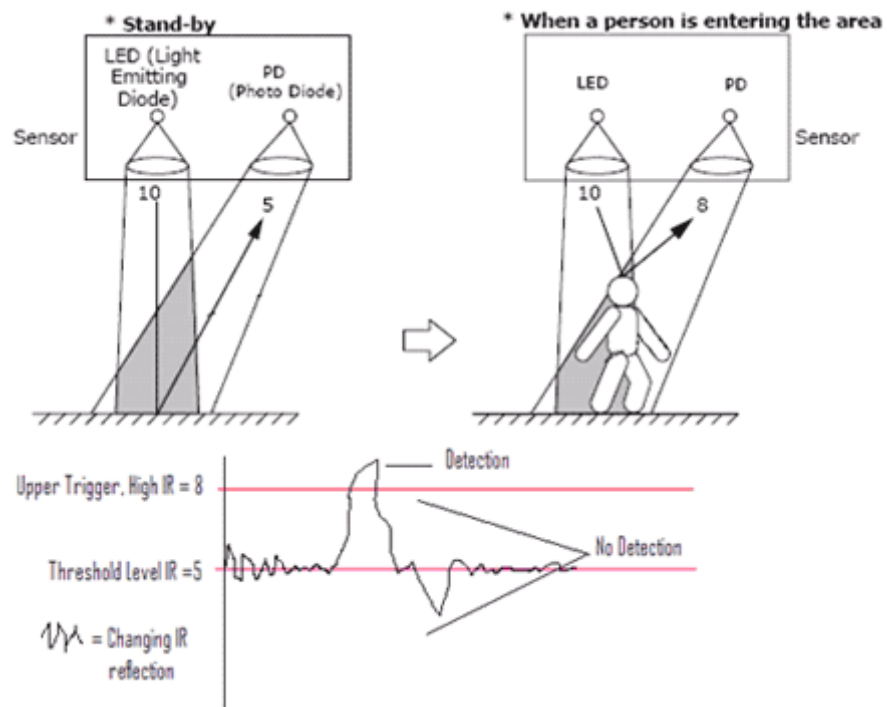


Рисунок 1.10– Принцип роботи інфрачервоного датчику

1.3.5 Магнітний датчик

Магнітний датчик (рисунок 1.11) - це датчик, який виявляє зміни в магнетизмі Землі та її периферійному магнітному середовищі на основі щільності магнітного потоку, використання цієї властивості дозволяє виявляти навколишні магнітні тіла. Крім того, датчик імпедансу магніту має більшу чутливість, ніж інші елементи, які також можуть вимірювати щільність магнітного потоку, такі як елемент Холла та елемент МР (магнітоопірний), та може виявляти рух магнітного тіла розташованого на висоті автомобіля над землею від датчика, встановленого на дорожньому покритті. [8]

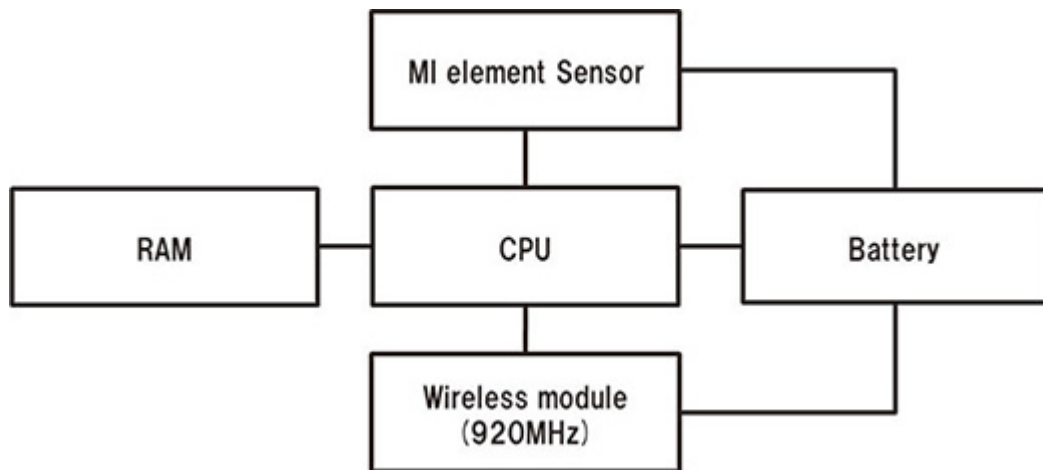


Рисунок 1.11– Принцип роботи магнітного датчику

1.4 Вибір методів для організації паркувального простору

Існуючі системи класифікуються за чотирма категоріями на основі методів виявлення:

- 1) системи на основі лічильників;
- 2) дротова система на основі датчиків;
- 3) на основі бездротового датчика;
- 4) системи на основі зображення або камери.

Системи на основі лічильників покладаються на датчики на вході та виїзді з автостоянок. Вони можуть надавати лише інформацію про загальну кількість вільних місць, а не направляти водіїв до точного розташування паркувальних місць, і такі системи не можна застосовувати до вуличних паркувальних місць та житлових паркувальних місць.

Провідні та бездротові системи на основі магнітних датчиків спираються на ультразвукове, інфрачервоне світло або бездротові магнітні датчики, встановлені на кожному паркувальному місці. Обидві системи були застосовані в практичному комерційному використанні, особливо в приміщенні, наприклад, в торгових центрах. Однак такі методи вимагають

встановлення дорогих датчиків (≈ 80 доларів США) на додаток до процесорів та трансиверів для бездротових технологій.

Системи на основі датчиків мають високий ступінь надійності, але їх висока вартість установки та обслуговування обмежує їх використання для широкого застосування. Порівняно з системами на основі датчиків, технології на основі камер відносно економічно ефективні, оскільки обидві функції загального спостереження та виявлення зайнятості автостоянки можуть виконуватися одночасно.

Розглянуті методи використовували також у різноманітних комбінаціях:

Існує система виявлення зайнятості місця для паркування з використанням комбінованих інфрачервоних датчиків з ультразвукових датчиків для зовнішнього/внутрішнього використання, щоб визначити, чи було зайняте місце для паркування. Ці датчики, як правило, надійні; проте недоліком методів на основі датчиків є значна вартість застосування технології на вулицях. Крім того, датчик паркувальних місць зазвичай застосовується на закритих автостоянках. Кожна сітка вимагає принаймні одного датчика та потребує серверного пристрою для обчислення паркувального пояса.

Таким чином, реалізація цього на вулицях є нездійсненною. Витрати на створення та обслуговування такої системи виявлення можуть безпосередньо вплинути на її якість, а встановлення сотень датчиків на великих або відкритих стоянках неможливе.

Методи на основі комп'ютерного зору знижують витрати та відображають зображення послідовності в режимі реального часу для спостереження за кількома об'єктами.

1.5 Висновок

У розділі було розглянуто існуючі платформи для вирішення проблеми паркувальних місць, що вже реалізовані у світі.

Проте для імплементації цих рішень у сучасні реалії потрібно виділити чимало грошей на потрібне обладнання.

Тож було вирішено створити свою систему, що буде використовувати вже наявні ресурси.

У розділі було також приведено опис можливих методів реалізації знаходження вільних місць, серед яких різноманітні види датчиків та відеокамери. Для розробки системи було обрано відеокамери.

Переваги використання систем на основі відеокамер порівняно з іншими існуючими системами:

По-перше, немає необхідності у додатковій інфраструктурі за умови, що об'єкт обладнаний камерами відеоспостереження, що закривають місця для паркування.

По-друге, системи на основі камер забезпечують точне розташування вільних паркувальних місць, що є вимогою для навігації транспортних засобів до вільних паркувальних місць.

По-третє, методи на основі камер дуже застосовні до вуличних та житлових паркувальних місць.

РОЗДІЛ 2 ЗАДАЧА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ З ВІДЕОПОТОКУ.

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ

2.1 Вступ

Виявлення об'єктів - це техніка комп'ютерного зору, за допомогою якої програмна система може виявляти, знаходити та відстежувати об'єкт із даного зображення чи відео. Спеціальним атрибутом виявлення об'єктів є те, що він ідентифікує клас об'єкта (особа, стіл, стілець тощо) та координати, характерні для їх розташування на даному зображенні. Місце розташування вказується шляхом обведення обмежувальної рамки навколо об'єкта. Обмежувальна коробка може або не може точно визначити положення об'єкта. Можливість розташування об'єкта всередині зображення визначає продуктивність алгоритму, який використовується для виявлення. Розпізнавання обличчя є одним із прикладів виявлення об'єктів.

Ці алгоритми виявлення об'єктів можуть бути попередньо навчені або їх можна навчити з нуля.

Комп'ютерний зір використовує різні дескриптори для виявлення об'єктів

2.2 Огляд дескрипторів

2.2.1 Локальні дескриптори

Локальні дескриптори описують фрагменти зображення (ключові точки на зображенні) об'єкта.

SIFT - масштабне інваріантне перетворення характеристик. [9]

В алгоритмі SIFT в основному є чотири кроки:

1. Виявлення екстремальних масштабів у просторі

Ми не можемо використовувати одне і те ж вікно для виявлення ключових точок різного масштабу. З маленьким куточком все нормально. Але для виявлення більших кутів нам потрібні більші вікна. Для цього використовується фільтрація масштабу-простору. У ньому знайдено Лапласіан Гаусса для зображення з різними значеннями σ . [10]

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

де L - значення гауссіана в точці з координатами (x, y) ,

σ - радіус розмиття,

G – ядро Гаусса,

I - значення вихідного зображення,

$*$ - операція згортки.

Лапласіан Гаусса діє як детектор крапель, який виявляє краплі різного розміру через зміну σ . Коротше кажучи, σ діє як параметр масштабування. Ядро Гаусса з низьким σ дає велике значення для маленького кута, тоді як ядро Гаусса з високим σ добре підходить для більшого кута. Отже, ми можемо знайти локальні максимуми у масштабі та просторі, що дає нам список значень (x, y, σ) , що означає, що є потенційна ключова точка у (x, y) на σ шкалі.

SIFT використовує різницю гаусіанів, яка є наближенням Лапласіана Гаусса. Різниця гаусіанів – це різниця гауссового розмиття зображення з двома різними σ , нехай це буде σ і $k\sigma$.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma),$$

Цей процес виконується для різних октав зображення в піраміді Гаусса. Він представлений на зображенні нижче (рисунок 2.1).

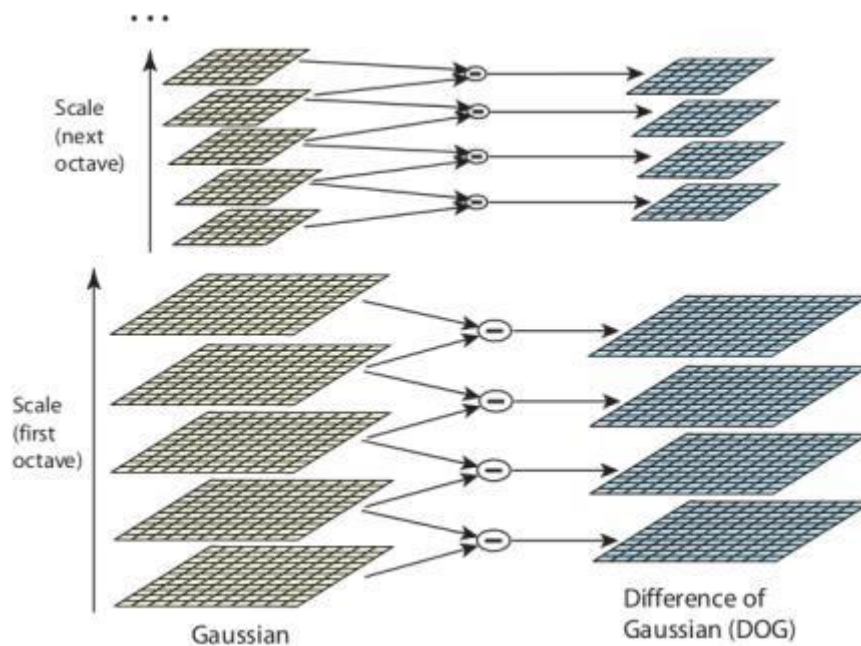


Рисунок 2.1– Масштабування за допомогою різниці гаусіанів

Після того, як ця DoG буде знайдена, зображення будуть шукати місцеві екстремуми у масштабі та просторі. Наприклад, один піксель зображення порівнюється з його 8 сусідами, а також з 9 пікселями у наступному масштабі та 9 пікселів у попередніх масштабах. Якщо це локальний екстремум, це потенційна ключова точка. В основному це означає, що ключова точка найкраще представлена в цій шкалі. Це показано на зображенні нижче (рисунок 2.2).

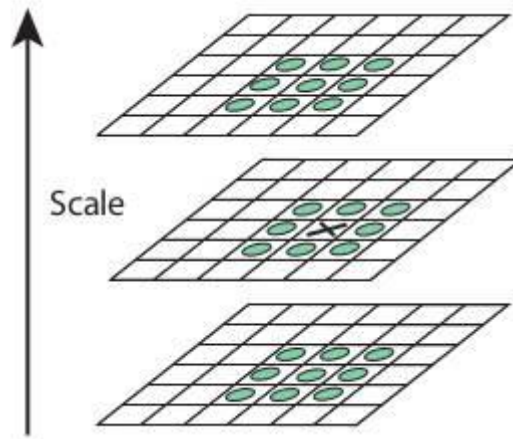


Рисунок 2.2– Пошук місцевих екстремумів

2. Локалізація ключових точок

Після того, як потенційні локації ключових точок будуть знайдені, їх потрібно уточнити, щоб отримати більш точні результати. Використовується розширення масштабного простору ряду Тейлора, щоб отримати більш точне місце розташування екстремумів, і якщо інтенсивність у цих екстремумах менше порогового значення, воно відхиляється.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x,$$

де D - функція DoG,

$X = (x, y, \sigma)$ - вектор зміщення відносно точки розкладання,

перша похідна DoG - градієнт,

друга похідна DoG - матриця Гессе

Екстремум многочлена Тейлора знаходиться шляхом обчислення похідної та прирівнювання її до нуля. В результаті отримаємо зсув точки обчисленого екстремуму, щодо точного.

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x},$$

Всі похідні обчислюються за формулами кінцевих різниць. У підсумку отримуємо СЛАР розмірності 3×3 , щодо компонент вектора $\hat{x}$.

Якщо одна з компонент вектора $\hat{x}$ більша за півкроку сітки в цьому напрямі, то це означає, що насправді точка екстремуму була обчислена невірно і потрібно зрушити до сусідньої точки в напрямку зазначених компонент. Для сусідньої точки все повторюється. Якщо таким чином ми вийшли за межі октави, то слід виключити дану точку з розгляду.

Коли стан точки екстремуму обчислено, перевіряється на малість саме значення DoG в цій точці за формулою

$$D(\hat{x}) = D + \frac{\partial D^T}{\partial x} \frac{1}{2} \hat{x}$$

Якщо ця перевірка не проходить, то точка виключається, як точка з малим контрастом.

DoG має вищу реакцію на краї, тому крайові також потрібно видалити. Для обчислення головної кривизни вони використовуються матрицю Гессе. Якщо цей коефіцієнт перевищує поріг, ця ключова точка відкидається.

$$H = \begin{bmatrix} D_{xx} & D_{yx} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

$$\alpha = r\beta,$$

де r – відношення більшого вигину до меншого,

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r},$$

Отже точка може розглядатись далі, якщо

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r + 1)^2}{r}.$$

Таким чином, усуваються будь-які низькоконтрастні ключові точки та крайові ключові точки.

3. Присвоєння орієнтації

Тепер кожній ключовій точці призначається орієнтація для досягнення інваріантності до обертання зображення. Навколо місця розташування ключової точки береться околиця в залежності від масштабу, і величина градієнта та напрямок обчислюються в цій області. Створюється орієнтаційна гістограма з 36 бункерами, що охоплюють 360 градусів (вона зважується за величиною градієнта та круговим вікном, зваженим за Гауссом, σ дорівнює 1,5-кратному масштабу ключової точки). У гістограмі береться найвищий пік, і будь-який пік вище 80% також розраховується для обчислення орієнтації. Він створює ключові точки з однаковим розташуванням та масштабом, але різними напрямками. Це сприяє стабільності відповідності.

4. Дескриптор ключової точки

Тепер створено дескриптор ключових точок. Береться район 16×16 навколо ключової точки. Він розділений на 16 субблоків розміром 4×4 . Для кожного підблоку створюється гістограма орієнтації на 8 bins. Таким чином, загалом доступно 128 значень контейнерів. Він представлений у вигляді вектора для формування дескриптора ключової точки. На додаток до цього, вживається кілька заходів для досягнення стійкості проти змін освітлення, обертання тощо.

5. Відповідність ключових точок

Ключові точки між двома зображеннями узгоджуються шляхом ідентифікації їх найближчих сусідів. Але в деяких випадках другий найближчий збіг може бути дуже близьким до першого. Це може статися через шум або інші причини. У цьому випадку береться відношення найближчої відстані до другої найближчої відстані. Якщо воно більше 0,8, вони відхиляються. Він усуває близько 90% помилкових збігів і відкидає лише 5% правильних, згідно з документом.

У SIFT Лапласіан Гаусса наблизили до різниці Гаусса для пошуку масштабу-простору, SURF йде трохи далі і наближає LoG з блочного фільтру (рисунок 2.3).

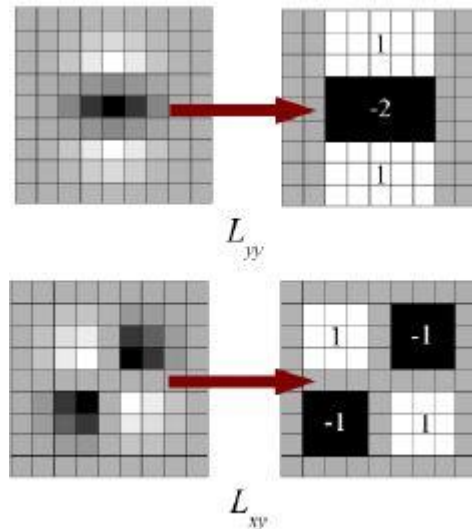


Рисунок 2.3– Приклад роботи блочного фільтру

Однією великою перевагою цього наближення є те, що згортку з блочним фільтром можна легко обчислити за допомогою інтегральних зображень. І це можна робити паралельно для різних масштабів. Крім того, SURF покладається на детермінант гесіанської матриці як для масштабу, так і для розташування. [10]

Для призначення орієнтації SURF використовує вейвлет-відповіді в горизонтальному та вертикальному напрямку для сусідства розміром 6s. До нього також застосовуються відповідні гауссові ваги. Потім вони розміщуються у просторі (рисунок 2.4).

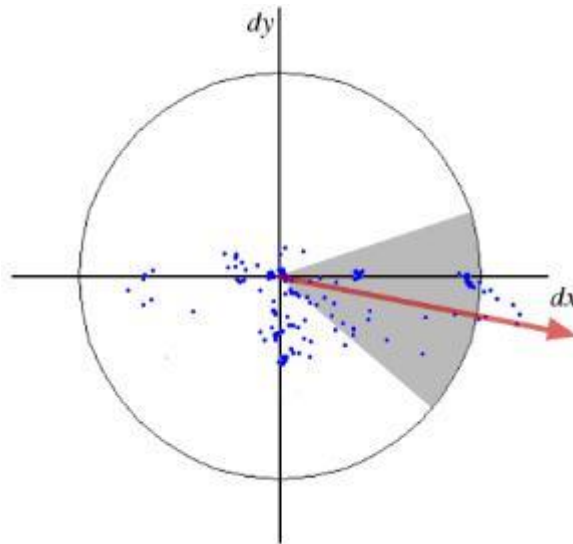


Рисунок 2.4— Гауссові ваги у просторі

Домінуючу орієнтацію оцінюють шляхом обчислення суми всіх відповідей у ковзному орієнтаційному вікні під кутом 60 градусів. Вейвлет-відповідь можна знайти за допомогою інтегральних зображень дуже легко в будь-якому масштабі. Для багатьох додатків інваріантність обертання не потрібна, тому нема потреби знаходити цю орієнтацію, що прискорює процес.

Іншим важливим поліпшенням є використання знака Лапласіана (сліду Гессіанської матриці) для базової точки інтересу. Це не додає витрат на обчислення, оскільки воно вже обчислюється під час виявлення. Знак Лапласіана відрізняє яскраві плями на темному тлі від зворотної ситуації. На етапі відповідності ми порівнюємо лише ті характеристики, які мають однаковий тип контрасту. Ця мінімальна інформація дозволяє швидше знайти відповідність без зниження продуктивності дескриптора.

SURF додає багато функцій для підвищення швидкості на кожному кроці. Аналіз показує, що він у 3 рази швидше, ніж SIFT, тоді як продуктивність порівнянна з SIFT. SURF добре обробляє зображення з розмиттям і обертанням, але погано справляється зі зміною точки огляду та освітленням.

SIFT - використовує 128-димний вектор для дескрипторів. Оскільки він використовує числа з плаваючою комою, він займає в основному 512 байт. Подібним чином SURF також займає мінімум 256 байт (для 64-dim). Створення такого вектора для тисяч функцій займає багато пам'яті, що неможливо для додатків з обмеженням ресурсів, особливо для вбудованих систем. Чим більше пам'ять, тим більше часу потрібно для збігу. [10]

Але всі ці розміри можуть не знадобитися для фактичного відповідності. Ми можемо стиснути його, використовуючи кілька методів, таких як PCA, LDA, що використовуються для перетворення цих дескрипторів SIFT у числа з плаваючою комою у двійкові рядки. Ці двійкові рядки використовуються для узгодження функцій за допомогою відстані Хеммінга. Це забезпечує кращу прискореність, оскільки для знаходження відстані відбиття потрібно лише застосувати XOR і підрахунок бітів. Але тут нам потрібно спочатку знайти дескриптори, потім тільки ми можемо застосувати хешування, що не вирішує нашу початкову проблему з пам'яттю.

У цей момент з'являється BRIEF. Він забезпечує ярлик для пошуку двійкових рядків безпосередньо без пошуку дескрипторів. Він бере згладжене виправлення зображення та вибирає набір пар розташувань (x, y) . Потім на цих парах розташування проводяться порівняння інтенсивності пікселів. Наприклад, нехай першими парами розташування є p і q . Якщо $I(p) < I(q)$, то його результат дорівнює 1, інакше - 0. Це застосовується до всіх пар n -го розташування, щоб отримати n -мірний рядок бітів.

Важливим моментом є те, що BRIEF - це дескриптор функцій, він не надає жодного методу пошуку функцій. Тому треба використовувати будь-які інші детектори функцій, такі як SIFT, SURF тощо.

BRIEF - це більш швидкий розрахунок і порівняння дескрипторів функціональних методів. Він також забезпечує високу швидкість розпізнавання, якщо немає великого обертання в площині.

2.2.2 Глобальні дескриптори

HOG, або гістограма орієнтованих градієнтів, - це дескриптор об'єктів, який часто використовується для вилучення об'єктів із даних зображення. На відміну від попередніх цей дескриптор є глобальним, такі дескриптори описують зображення в цілому для узагальнення всього об'єкта. Глобальні функції включають уявлення контурів, дескриптори форми та особливості текстури. HOG широко використовується в задачах комп'ютерного зору для виявлення об'єктів. [10]

Дескриптор HOG зосереджується на структурі або формі об'єкта. У разі крайових особливостей ми визначаємо лише, чи є піксель краєм, чи ні. HOG також може забезпечити напрямок краю. Це робиться шляхом вилучення градієнта та орієнтації (або можна сказати величини та напрямку) країв.

Крім того, ці орієнтації обчислюються "локалізованими" частинами. Це означає, що повне зображення розбивається на менші області, і для кожної області розраховуються градієнти та орієнтація.

Тоді HOG створить гістограму для кожної з цих областей окремо. Гістограми створюються з використанням градієнтів та орієнтацій значень пікселів, звідси назва «Гістограма орієнтованих градієнтів» (рисунок 2.5).

Отже, дескриптор функції HOG підраховує випадки орієнтації градієнта в локалізованих частинах зображення.

Крок 1. Попередня обробка даних

Попередня обробка даних є вирішальним кроком у будь-якому проєкті машинного навчання, і це не відрізняється при роботі з зображеннями.

Потрібно попередньо обробити зображення і зменшити співвідношення ширини до висоти до 1:2. Переважно розмір зображення повинен бути 64 x 128. Це тому, що ми будемо розділяти зображення на патчі 8*8 та 16*16 для вилучення об'єктів. Наявність зазначеного розміру (64 x 128) зробить усі наші розрахунки досить простими.

Крок 2: Обчислення градієнтів (напрямок x та y)

Наступний крок - обчислення градієнта для кожного пікселя зображення.

Виділяємо значення одного пікселя. Тепер, щоб визначити градієнт (або зміну) у напрямку x, нам потрібно відняти значення зліва від значення пікселя справа. Аналогічно, щоб обчислити градієнт у напрямку y, ми віднімемо значення пікселя нижче від значення пікселя над вибраним пікселем.

Цей процес дасть нам дві нові матриці: одна зберігає градієнти в напрямку x, а інша — у напрямку y. Це схоже на використання ядра Собеля розміром 1. Величина буде вищою при різкій зміні інтенсивності, наприклад, по краях.

Цей процес повторюється для всіх пікселів на зображенні. Наступним кроком буде визначення величини та орієнтації за допомогою цих значень.

Крок 3: Обчислити величину та орієнтацію

Використовуючи градієнти, визначається величину та напрямок для кожного значення пікселя.

Далі потрібно створити гістограму, використовуючи ці градієнти та орієнтації.

Крок 4: Обчислити гістограму градієнтів у клітинках 8×8

Гістограми, створені в дескрипторі функцій HOG, не генеруються для всього зображення. Натомість зображення поділяється на комірки розміром 8×8 , і гістограма орієнтованих градієнтів обчислюється для кожної клітинки.

Таким чином ми отримуємо функції (або гістограму) для менших ділянок, які, у свою чергу, представляють ціле зображення..

Якщо ми розділимо зображення на осередки 8×8 і сформуємо гістограми, ми отримаємо матрицю 9×1 для кожної клітинки.

Ми візьмемо значення кожного пікселя, знайдемо величину градієнта пікселя та оновимо таблицю частот, що позначає градієнт та появу на зображенні.

При цьому, більша вага повинна бути у значення комірки, яка ближче до орієнтації.

Після того, як ми створили HOG для патчів 8×8 на зображенні, наступним кроком є нормалізація гістограм.

Крок 5: Нормалізація градієнтів у комірці 16×16 (36×1)

Градієнти зображення чутливі до загального освітлення. Це означає, що для певного зображення деяка частина зображення буде дуже яскравою порівняно з іншими частинами. Ми можемо зменшити цю зміну освітлення, нормалізувавши градієнти, взявши блоки 16×16 .

Отже, у нас буде чотири матриці 9×1 або одна матриця 36×1 . Щоб нормалізувати цю матрицю, ми розділимо кожне з цих значень на квадратний корінь із суми квадратів значень (рисунок 2.5).

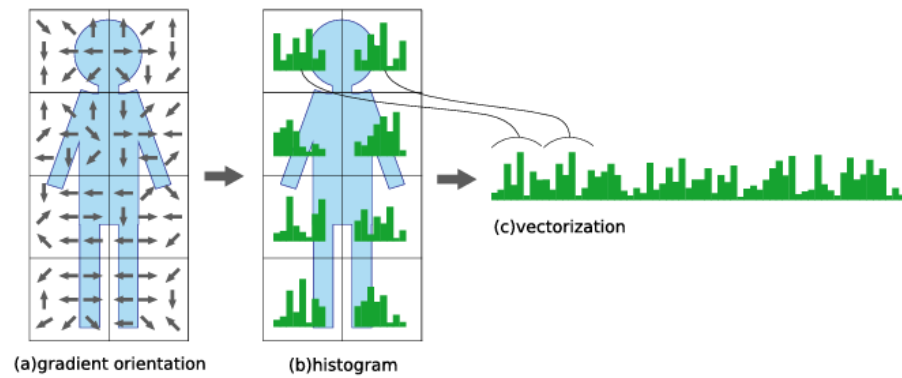


Рисунок 2.5– Алгоритм роботи HOG

HOG має дві переваги для виявлення об'єктів. Однією з переваг є стійкість проти дисперсії освітленості, оскільки орієнтації градієнта обчислюються на основі місцевої різниці інтенсивності. А також, стійкість проти деформацій, оскільки незначні зсуви та афінні деформації вносять невеликі зміни у значення гістограм.

Co-HOG - гістограми спільного виникнення орієнтованих градієнтів (функція використовує пари орієнтацій градієнта як одиниці, з яких вона будує гістограми). Матриця спільного виникнення (гістограма) виражає розподіл орієнтацій градієнта при заданому зміщенню над зображенням (рисунок 2.6).

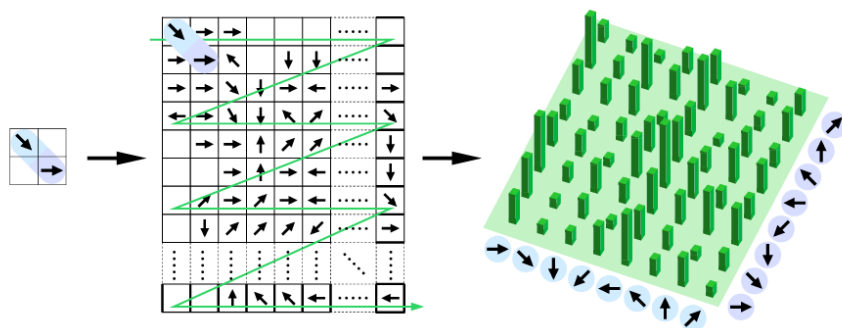


Рисунок 2.6– Матриця спільного виникнення

Комбінації сусідніх орієнтацій градієнта можуть детально виражати форми. Тож він є інформативним для класифікації об'єктів під кутом. Математично матриця спільного входження визначається над $n \times m$ зображенням, параметризованим зміщенням (x, y) , як:

$$C_{x,y}(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & I(p, q) = i \text{ та } I(p + x, q + y) = j \\ 0, & \text{інакше} \end{cases}, \quad (1)$$

де C - матриця спільного входження,
 n, m – розмір зображення у пікселях,
 x, y – зміщення,
 i, j - орієнтацій градієнта.

Отже, алгоритм підрахунку гістограми наступний:

```

1: given I: an image of gradient orientation
2: initialize H  $\leftarrow$  0
3: for all positions (p,q) inside of the image do
4:   i  $\leftarrow$  I(p,q)
5:   k  $\leftarrow$  the small region including (p,q)
6:   for all offsets (x,y) such that corresponds neighbors do
7:     if (p +x,q +y) is inside of the image then
8:       j  $\leftarrow$  I(p+x, q+y)
9:       H(k, i, j, x, y)  $\leftarrow$  H(k, i, j, x, y) + 1
10:    end if
11:  end for
12: end for

```

По-перше, ми обчислюємо орієнтації градієнта на зображенні

$$\theta = \arctg \frac{v}{h}, \quad (2)$$

де v і h - вертикальний та горизонтальний градієнти відповідно, обчислені за допомогою фільтра Собеля, фільтра Робертса тощо.

Ми позначаємо кожен піксель з однією з восьми дискретних орієнтацій або як безградієнтний. Усі орієнтації $0^\circ - 360^\circ$ поділені на вісім орієнтацій через 45° . Без градієнта означає, що $\sqrt{h^2 + v^2}$ менший за поріг.

По-друге, ми обчислюємо матриці спільного входження за рівнянням (1). Використовуючи короткі та дальні зміщення, матриця спільного виникнення може виражати локальні та глобальні форми. Ми не використовуємо половину зміщень, тому що вони ведуть себе так само, як і інші при обчисленні матриці спільного виявлення. Матриці спільного виникнення обчислюються для кожної невеликої області. Невеликі прямокутні області розміщені плиткою $N \times M$ без перекриття.

Нарешті, компоненти всіх матриць спільного появи об'єднуються у вектор (рисунок 2.7). [10]

Оскільки CoHOG виражає форми в деталях, він є високорозмірним. Матриця спільного виникнення, розрахована з нульовим зміщенням, має лише вісім ефективних значень, оскільки недіагональні компоненти дорівнюють нулю. Таким чином, CoHOG отримує тисячі компонентів із зображення. Насправді ефективних значень менше, оскільки матриці спільного повторення мають кілька компонентів з нульовим значенням. Компоненти з нульовим значенням не використовуються в класифікації,

оскільки їх внутрішній добуток завжди дорівнює нулю. Тим не менш, CoHOG є більш потужним дескриптором функцій, ніж HOG. Реалізація CoHOG проста. Ми можемо обчислити CoHOG лише шляхом ітерації для збільшення компонентів матриць спільного появи, тоді як розрахунок HOG включає більше процедури, такі як зважене голосування за орієнтацією, нормалізація гістограми, перекриття регіонів тощо.

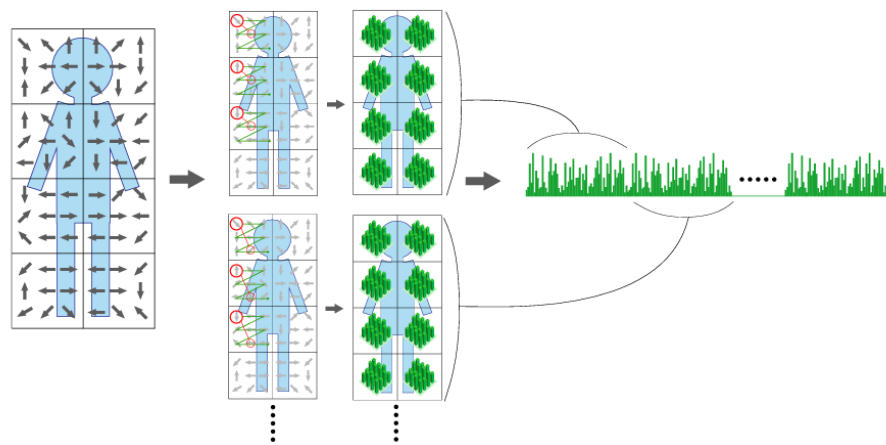


Рисунок 2.7– Алгоритм роботи Co-HOG

CoHOG має стійкість проти деформації та відхилення освітленості з тих же причин, що і HOG, оскільки CoHOG - це опис характеристик гістограми на основі градієнта.

2.3 Розпізнавання вільних місць на паркувальній зоні з відеопотоку

Існують різні підходи до виявлення зайнятості автостоянок.

2.3.1 Аналіз основних компонентів

Використовують алгоритм порівняння еталонного зображення та наборів вхідних даних для розрахунку площі пікселів автомобіля до місця паркування, використовуючи аналіз основних компонентів. [11]

Основна ідея аналізу основних компонентів ядра полягає в тому, щоб неявно зіставити дані у вхідному просторі з простором ознак за допомогою функції ядра, а потім обробити аналіз головних компонентів ядра у просторі функцій. Це потужний підхід до вилучення нелінійних ознак, реалізації зменшення параметрів ознак, зменшення надмірності даних та отримання первинних векторів ознак з низьким рівнем помилок класифікації та високою чутливістю до умов водіння.

2.3.2 Навчання байєсівського класифікатора

Навчання байєсівського класифікатора для виявлення транспортних засобів з використанням кутів, країв та вейвлетів. [11]

По-перше, представлено перетворення кольору для проєктування всіх кольорів вхідних пікселів у колірному просторі, щоб пікселі транспортного засобу могли бути зосереджені на меншій площі. Потім пропонується

байєсівська мережа для визначення кольорів транспортних засобів на фоні. Оскільки транспортні засоби мають різні розміри та орієнтацію, з кожного виявленого пікселя генерується кілька гіпотез транспортного засобу. Для перевірки кожної гіпотези ми використовуємо два види транспортних засобів, щоб відфільтрувати неможливих кандидатів на транспортні засоби. Особливості включають ребра та коефіцієнти вейвлет-перетворення. Тоді бажані транспортні засоби можна дуже надійно та точно перевірити та виявити на статичних зображеннях (рисунок 2.8).

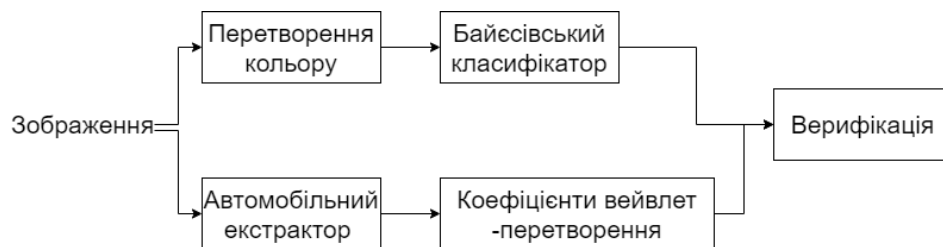


Рисунок 2.8– Алгоритм роботи системи на основі байєсівського класифікатора

2.3.3 Класифікації гістограм кольору

Перший крок у запропонованій системі VPSD - це захопити нерухоме, зображення стоянки та вилучити окремі місця для паркування. Використовуючи мітки ROI для місць паркування, система витягує інформацію з кожного окремого місця для паркування та передає її до класифікатора кольорової гістограми та алгоритму виявлення функціональних точок. [12]

Класифікатор кольорової гістограми (рисунок 2.9) перетворює колірний простір з RGB в $L^*a^*b^*$. Відкидаючи канал яскравості (L^*) і

зберігаючи лише канали кольоровості (a^*b^*), зображення стають (переважно) інваріантними до освітлення. Після цього для кожного додаткового зображення паркувального місця будується кольорова гістограма - 32 bins для каналу a^* і 32 bins для каналу b^* . Ці кольорові гістограми потім класифікуються або за допомогою машини опорних векторів, або за допомогою класифікатора k -найближчих сусідів.

Класифікатор k -найближчого сусіда -це простий алгоритм класифікації, з високою ефективністю. Час виконання класифікації є дуже повільним і продовжує сповільнюватися зі збільшенням розміру навчального набору. Однак важливо зазначити, що коли розмір навчального набору наближається до нескінченності, рівень помилок k -NN гарантовано буде не гірше, ніж подвійна частота помилок Байєса (мінімально досяжний рівень помилок). Алгоритм працює шляхом порівняння даних, які потрібно класифікувати, зі всіма навчальними даними. Відстань між вхідними даними та кожним екземпляром навчальних даних обчислюється і записуються найближчі навчальні точки. Точки даних потім голосують, а вхідні дані позначаються найчастішим класом. Щоб запобігти збігу в процесі голосування, зазвичай їх непарне число. Існує багато різних показників відстані, які може використовуватися класифікатором k -NN. Найбільш відома функція відстані - це Евклідова метрика відстані.



Рисунок 2.9– Зображення роботи системи на основі класифікації гістограм кольору

2.3.4 Навчання класифікатора SVM на основі текстур

Навчання класифікатора SVM щодо багатьох текстурних особливостей та покращення ефективності виявлення за допомогою ансамблів SVM. [12]

Спочатку проводиться попередня обробка:

1. Вилучення функції із зображення.

У зображенні є багато функцій, які можна витягти та допомогти у навчанні класифікатора. Представлення зображення шляхом вилучення з нього лише корисної інформації відоме як дескриптор функцій. Для цього методу найчастіше використовується HOG.

Виділення просторових ознак зображення:

Змінюємо розмір зображення, зберігаючи всі його особливості, оскільки за допомогою цього код буде працювати трохи швидше та без втрати інформації.

2. Перетворення кольорового простору зображення.

Проводиться перетворення зображення в певний колірний простір для вилучення інформації з вибраного колірного каналу.

Зображення можна конвертувати в будь-яке з наведених нижче кольорових просторів:

RGB, HSV, YCrCb, YUV, LUV, HLS

Наприклад, якщо потрібно розпізнати зображення об'єкта під деякою тінню, можна скористатися деякими їх властивостями, такими як легкість або насиченість зображення, щоб більш ефективно класифікувати транспортні засоби на деяких тінях. Це можна зробити за допомогою каналу колірного простору HLS.

3. Вилучення кольорової гистограми.

4. HOG.

HOG - це останній крок, який використовується у процесі вилучення функцій.

Після того, як видобуток об'єктів буде завершено, починається навчання класифікатора.

Як показує практика, для навчання (Рисунок 2.10) краще використовувати rbf SVM (функція радіального базису в машині підтримки векторів). З лінійною SVM знаходиться багато областей, де виявлення транспортних засобів є неможливим через меншу точність.

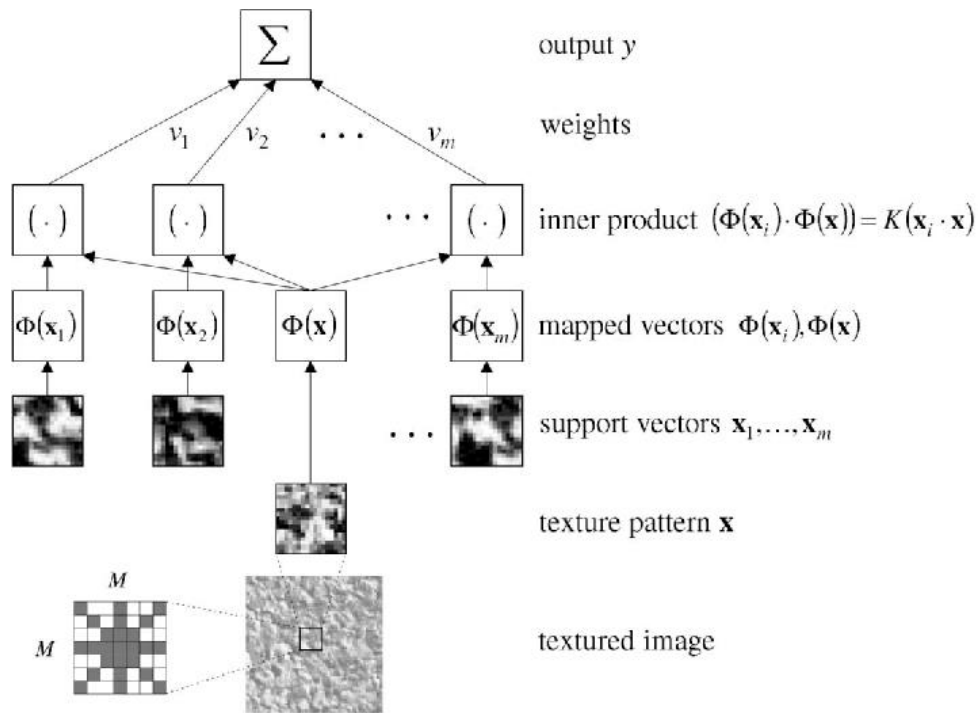


Рисунок 2.10– Навчання класифікатора SVM на основі текстур

Згадані вище методи ґрунтуються на особливостях, створених вручну (таких як краї, колір, текстура) та без урахування фону, що робить ці методи сприйнятливими до різних погодних умов та змін освітлення.

2.3.5 CNN

CNN - це алгоритм машинного навчання, який використовує локальні просторові дані в зображенні та вивчає ієрархію дедалі складніших функцій, тим самим автоматизуючи процес побудови об'єктів. Рамки на основі CNN досягли найкращої точності в класифікації зображень та виявленні об'єктів.

У літературі є чіткі докази того, що навчання за допомогою CNN перевершує традиційні методи з використанням функцій ручної роботи для

виявлення статусу автостоянок. Та майже всі системи на базі CNN тонко налаштовують готові попередньо навчені мережі.

Традиційні методи обробки зображень виявилися успішними в експериментальних і реальних ситуаціях. Проблемою традиційних методів обробки зображень, як правило, є вилучення функцій. Розробка функцій для моделі автомобіля може бути складною, оскільки існує багато типів автомобілів, і мати узагальнену модель буде важко. З цієї точки зору складні ознаки можна вивчити через CNN архітектури, такі як AlexNet, або моделі виявлення об'єктів, такі як YOLO. Єдиним недоліком CNN є те, що CNN дорогі в обчислювальному плані під час навчання, але швидкі у висновках. Щоб CNN працювала найкраще, потрібен великий набір даних, що містить усі можливі ситуації того, що підлягає класифікації.

2.3.5.1 VGGNet-F

Існує рішення навколо CNN VGGNet-F:

Запропонована система складається з трьох частин (рисунок 2.11).

По-перше, це візуальні вузли, а саме камери, які підключені до сервера через локальну бездротову мережу або через Інтернет. Сервер - другий компонент. Він включає базу даних, модуль виявлення, веб-службу та обробник подій. Сервер збирає зображення з візуальних вузлів, передає їх до модуля виявлення, збирає та зберігає вихідні дані в базі даних і надає веб-сервіс для інтерфейсних програм для отримання інформації з бази даних. Третій елемент – це інтерфейс, який представляє користувачам вільні місця для паркування. [13]

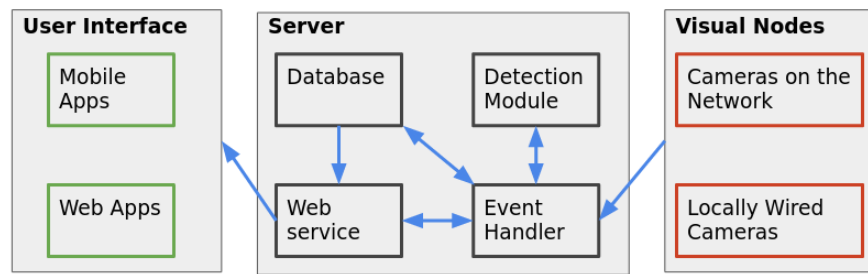


Рисунок 2.11– Схема роботи системи розпізнавання на базі CNN VGGNet-F

У цьому проєкті використовуються звичайні кольорові камери та реляційна база даних. Сервер зберігає окремі таблиці для різних стоянок, де кожен елемент таблиці відповідає паркувальному місцю. Кожне місце має чотири обов'язкові поля.

1. Унікальний номер на місця.
2. Координати обмежувальних ліній (у просторі зображення). Координати вводяться через графічний інтерфейс адміністратором.
3. Зображення. Обрізане зображення стійла з поточної візуальної стрічки. Це буде оновлюватися сервером через фіксовані проміжки часу.
4. Статус. Двійкове значення, яке вказує на вакантність або зайнятість. Це також оновлюється сервером із відповіді методу виявлення.

По-друге, це збір даних з камер. Камери можуть бути підключені до сервера в локальній мережі або через Інтернет. Якщо камери не використовують протокол HTTP, на сервері буде використовуватися локальний протокол зв'язку. Якщо вони це роблять, запити та відповіді можна виконувати за допомогою HTTP. Якщо камери підключені до Інтернету, немає потреби, щоб сервер знаходився поблизу візуального вузла.

По-третє, обслуговування веб-сервіса. Він з'єднує базу даних з інтерфейсом нашої системи. Основний підтримуваний функціонал - це отримання статусу всіх місць на кожній автостоянці.

Нарешті, він подає зображення з візуальних вузлів до модуля виявлення разом з обмежувальними коробками стійлів і отримує передбачення модуля виявлення.

Модуль виявлення відповідає за звіт про стан зайнятості паркувального місця, враховуючи зображення. Для цього завдання використано згорткову нейронну мережу.

Наявність кількох обмежень на вхідні дані ставить важкий тягар на систему виявлення. Тому надійність та загальність алгоритму виявлення мають найвищий пріоритет. Для цього вирішальне значення має правильний вибір мережі та навчальної процедури. Дизайн нашої мережі базується на VGGNet-F (рисунок 2.12). Він має п'ять згорткових шарів, за кожним слідує шар об'єднання та функція активації виправленої лінійної лінії. Він має три повністю з'єднані шари на кінці, які використовують ознаки згорткових шарів для класифікації.

Архітектура VGGNet відрізняється простим і єдиним дизайном по всій мережі.

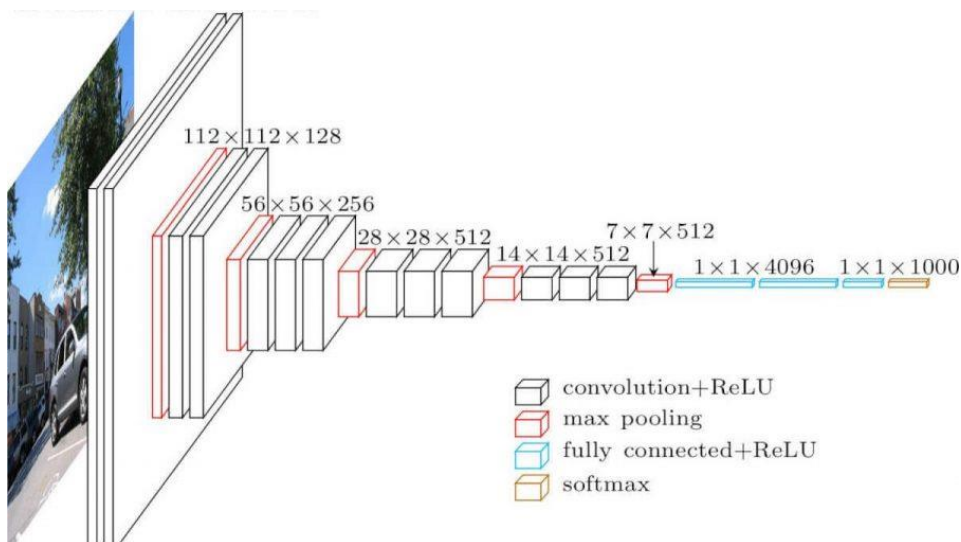


Рисунок 2.12– Шари системи розпізнавання на базі CNN VGGNet-F

Розмір ядра фільтрів – 3×3 , крім перших двох (11 і 5), щоб зменшити розмір мережі.

На вхід шару conv1 подаються RGB зображення розміру 224x224. Далі зображення проходять через стек згортальних шарів, в яких використовуються фільтри з дуже маленьким рецептивним полем розміру 3x3 (який є найменшим розміром для отримання уявлення про те, де знаходиться право / ліво, верх / низ, центр).

В одній з конфігурацій використовується згортковий фільтр розміру 1x1, який може бути представлений як лінійна трансформація вхідних каналів (з подальшою нелінійністю). Згортковий крок фіксується на значенні 1 піксель. Просторове доповнення (padding) входу згорткового шару вибирається таким чином, щоб просторова роздільна здатність зберігалася після згортки, тобто доповнення дорівнює 1 для 3x3 згортальних шарів. Просторовий пулінг здійснюється за допомогою п'яти max-pooling шарів, які слідує за одним з згортальних шарів (не всі згорткові шари мають наступні max-pooling). Операція max-pooling виконується на вікні розміру 2x2 пікселів з кроком 2.

Після стека згортальних шарів (який має різну глибину в різних архітектурах) йдуть три повнозв'язних шари: перші два мають по 4096 каналів, третій - 1000 каналів (так як в змаганні ILSVRC потрібно класифікувати об'єкти по 1000 категоріям; отже, кожному класу відповідає один канал). Останнім йде soft-max шар.

Всі приховані шари забезпечені ReLU.

Існує декілька рішень з модифікаціями для задачі розпізнавання вільного паркувального місця:

- Мережі (за винятком однієї) не містять шару нормалізації (Local Response Normalisation), так як нормалізація не покращує результату на датасетах, а веде до збільшення споживання пам'яті та часу виконання коду.

- Останній повністю підключений шар змінили для виведення двійкових значень для зайнятого або порожнього місця. Це зменшило кількість параметрів мережі та зменшило її ефективний розмір. Модуль виявлення може бути розміщений всередині сервера, або він може працювати

у хмарі і спілкуватися лише з сервером. Така реалізація досягла точності 99,99%.

2.3.5.2 AlexNet

Архітектура мережі AlexNet (рисунок 2.13) включає в себе 8 шарів з ваговими коефіцієнтами. Пеший-п'ятий - згорткові, інші - повнозв'язні. Далі дані проходять через функцію втрат softmax, яка визначає 1000 міток різних класів. Мережа максимізує багатолінійну логістичну регресію, що еквівалентно максимізації середнього по всім навчальним випадкам логарифма ймовірності правильного маркування з розподілу очікування. Ядра 2, 4 і 5 згорткових шарів мають зв'язок тільки з тими картами в попередньому шарі, які розташовуються на одному графічному процесорі. Ядра 3 згорткового шару зв'язані з усіма картами ядер 2 шару. Нейрони в повнозв'язних шарах зв'язані з усіма нейронами попереднього шару. [14]

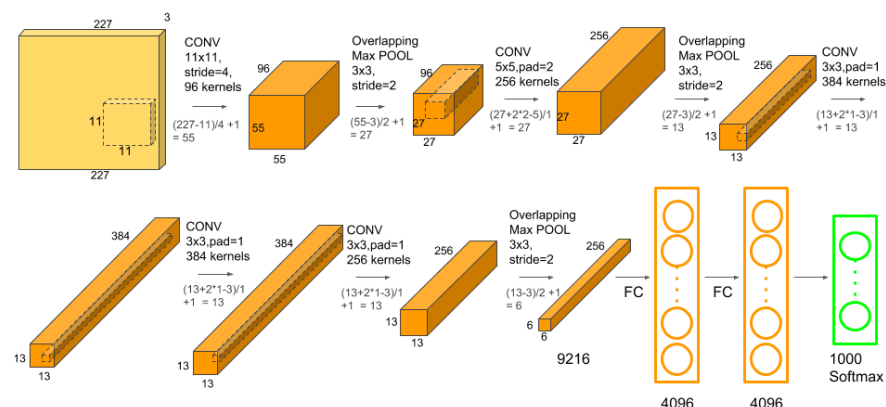


Рисунок 2.13– Шари системи розпізнавання на базі AlexNet

Отже, AlexNet складається з 5 згорткових шарів і 3 повнозв'язних шарів. Relu використовується після кожного з них. Дропаут використовуємо перед I і II повнозв'язними шарами. Мережа включає в себе 62300000 параметрів і виконує 1100000000 обчислень при прямому проході. Згорткові шари, які використовують всього 6% параметрів, виконують 95% обчислень.

Для вирішення проблеми виявлення вільних паркувальних місць використовують модифіковану версію архітектури. Щоб запустити його на Raspberry Pi, вони різко зменшили кількість згорткових шарів, щоб усі обчислення відбувалися в R.Pi, який дуже обмежений з точки зору ресурсів, ємності оперативної пам'яті та дискової пам'яті. Була незначна втрата точності між мережами, які вони модифікували, та їхньою власною. Їх модифікований CNN вже пройшов навчання (ваги вже встановлені), R.Pi's виконує класифікацію, яка є єдиним прямим проходом через мережу. Цей підхід особливий тим, що замість того, щоб створити сервер, який оброблятиме кадри з парковки та надсилатиме їх клієнтам, було вирішено створити свої розумні камери, кожна з яких складається з камери Raspberry Pi та Pi. Raspberry Pi виконає всі обчислення на борту та поверне результати на сервер.

2.3.5.1 R-CNN

R-CNN – згорткова нейронна мережа на основі регіонів, яка використовується для завдань комп'ютерного зору, зокрема для виявлення об'єктів.

Цей підхід використовує обмежувальні рамки між областями об'єкта, які потім оцінюють згорткові мережі незалежно від усіх регіонів інтересу (ROI), щоб класифікувати кілька областей зображення до запропонованого класу.

Fast R-CNN – це покращена версія архітектури R-CNN, що застосовує мережу регіональних пропозицій (RPN). RPN — це просто нейронна мережа, яка пропонує кілька об'єктів, доступних у певному зображенні. Вона витягує функції, використовуючи RoIPool (об'єднання регіонів інтересів), з кожного поля-кандидата і виконує класифікацію та регресію обмеженої рамки. RoIPool — це операція для вилучення невеликої карти об'єктів з кожного RoI під час виявлення.

«Fast R-CNN» швидше, ніж R-CNN, тому що вам не потрібно щоразу передавати 2000 пропозицій регіонів у згорткову нейронну мережу. Натомість операція згортки виконується лише один раз для кожного зображення, і з нього створюється карта об'єктів. Faster R-CNN – ще одна модифікація, яка використовує мережу регіональних пропозицій (RPN) та архітектури Fast R-CNN, вона була створена для ще більшого підвищення швидкості обчислень.

Mask R-CNN (рисунок 2.14) була створена за допомогою Faster R-CNN. У той час як Faster R-CNN має 2 виходи для кожного об'єкта-кандидата, мітку класу та зміщення обмежувальної рамки, Mask R-CNN додає третю гілку, яка виводить маску об'єкта. Вивід додаткової маски відрізняється від виводів класів і блоків, що вимагає вилучення набагато більш тонкого просторового макета об'єкта. Mask R-CNN є розширенням Faster R-CNN і працює шляхом додавання гілки для передбачення маски об'єкта (область інтересу) паралельно з існуючою гілкою для розпізнавання обмежувального прямокутника.

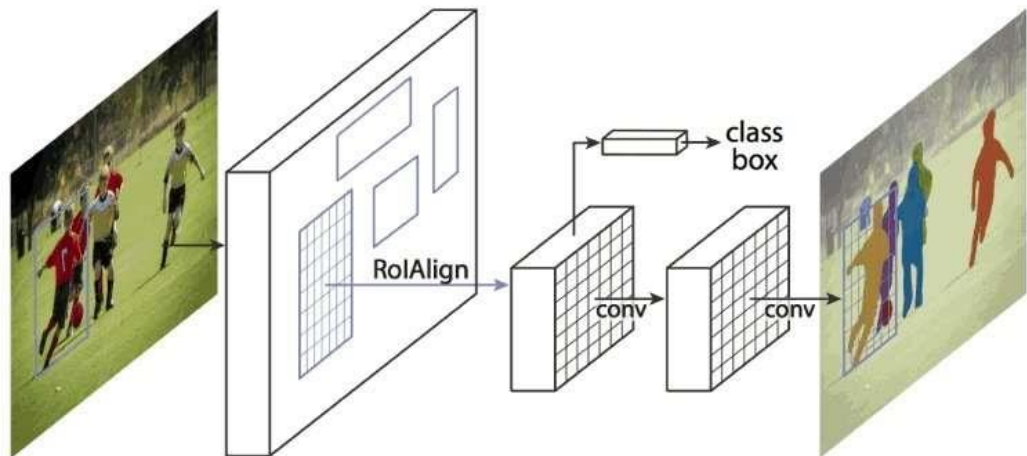


Рисунок 2.14– Архітектура Mask R-CNN

2.4 Набори даних

2.4.1 PKLot

Набір даних PKLot містить 12417 зображень 3 паркувальних зон (рисунок 2.15), з яких було створено та позначено у пакеті даних 695899 сегментованих паркувальних місць. Зображення було зроблено з 5-хвилинним інтервалом часу протягом 30 днів протягом дня за трьох погодних умов, а саме дощових, сонячних та похмурих днів. Зображення зроблені з різних місць і орієнтацій, охоплюючи транспортні засоби під різними кутами і розмірами. Кількість зайнятих та порожніх місць для паркування становить приблизно однаковий відсоток усього набору даних PKLot - відповідно 48,54% та 51,46%.

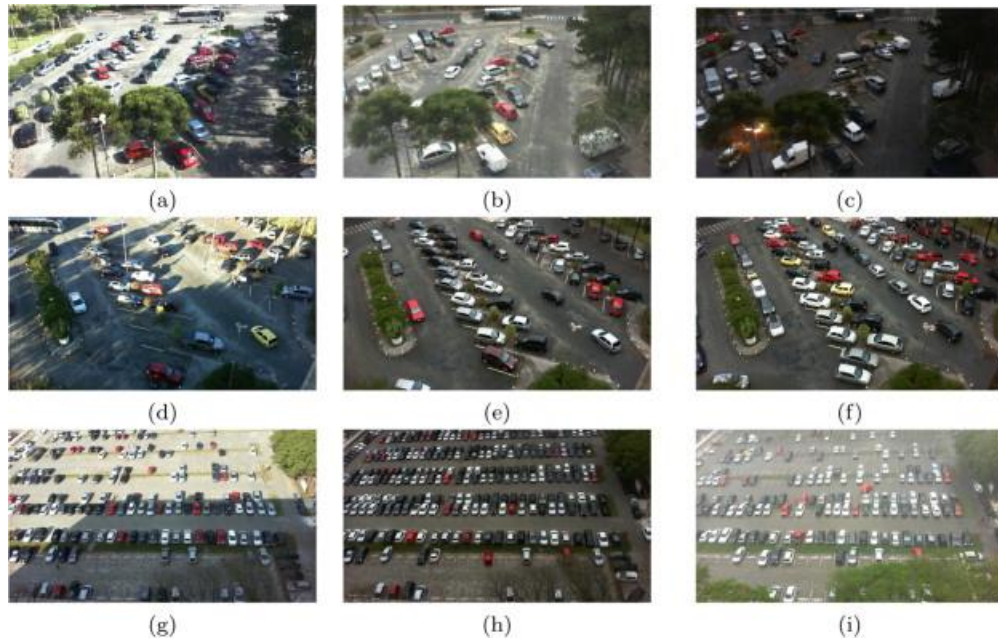


Рисунок 2.15– Зображення PKLot

2.4.2 CNRPark + EXT

Набір даних містить майже 150 000 маркованих зображень порожніх і використаних паркувальних місць (рисунок 2.16). Сет містить фотографії автостоянок, отримані в різні дні з листопада 2015 року по лютий 2016 року за різних погодних умов за допомогою 9 камер з різними кутами огляду.

У цьому наборі даних задіяні часткові моделі оклюзії через перешкоди, такі як ліхтарні стовпи, дерева, інші автомобілі та різні умови освітлення. Це дозволяє навчити класифікатор, який здатний охарактеризувати більшість складних обставин, які можуть бути виявлені в реальному житті.

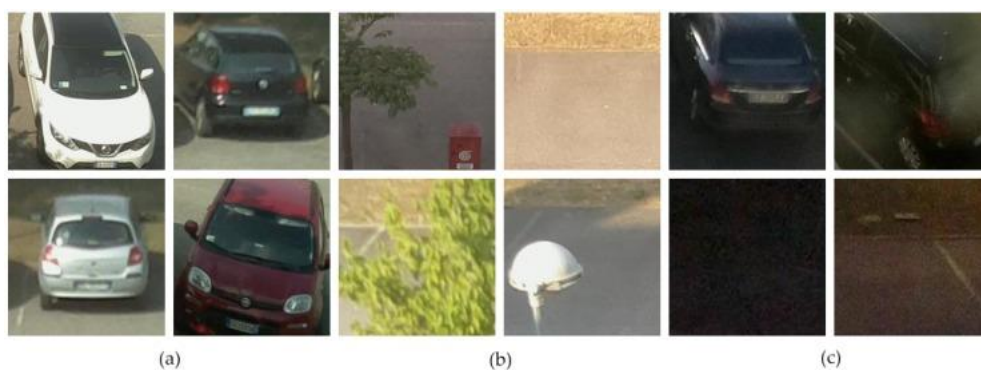


Рисунок 2.16— Зображення CNRPark + EXT

У PKLot зображення отримують із застосуванням обернутих прямокутних масок. На відміну від нього, у наборі даних CNRPark + EXT місця для паркування здебільшого не включають виокремлені місця для паркування. Крім того, CNRPark + EXT містить сильно закриті простори, такі як майже повністю приховані деревами або стовпами для ліхтарів. Однак ці перешкоди є менш впливовими у наборі сегментованих просторів набору даних PKLot. Крім того, знімки були зроблені з нижчого рівня зору, ніж PKLot, забезпечуючи додаткові закупорки через наступні автомобілі. Однією з дуже важливих особливостей набору даних CNRPark + EXT є те, що він також містить більш темні та нічні зображення. Це допомагає зробити навчальні образи різноманітними та створити середовище, подібне до сценаріїв із реального життя.

Кількість даних можна порівняти у таблиці 2.1.

Таблиця 2.1 –Порівняння наборів даних PKLot та CNRPark + EXT

Набір даних	Вільні місця	Зайняті місця	Всього
PKLot	337,780	358,119	695,899
CNRPark + EXT	65,658	79,307	144,965

2.5 Висновки

Ознайомившись з методами виявлення об'єктів з видеопотоку та практичними успіхами існуючих рішень задачі розпізнавання, було визначено, що найефективнішим методом на сьогоднішній день є згорткові нейронні мережі, які дають доволі високий відсоток точності.

AlexNet має близько 60 млн параметрів.

VGGNet не тільки має більшу кількість параметрів і FLOP в порівнянні з AlexNet, але також має меншу точність. Для навчання VGGNet з меншою точністю потрібно більше часу.

Навчання AlexNet займає приблизно той самий час, що й підготовка до початку навчання. Вимоги до пам'яті зменшуються в 10 разів із покращеною точністю (близько 9%).

РОЗДІЛ 3 ОПИС АРХІТЕКТУРИ ТА ПРОГРАМНОГО ПРОДУКТУ

3.1 Визначення кордонів паркувальних місць

Для виконання програми застосовуємо відео з відеокамери, що має наступний вигляд (рисунок 3.1).



Рисунок 3.1– Зображення відеокамери

Задача полягає в тому, щоб обробити зображення та визначити стан паркувальних місць на ньому.

По-перше, потрібно визначити кордони паркувальних місць на зображенні.

Більшість локальних рішень пошуку вільних місць за допомогою відеокамери використовують метод ручного програмування кожного поля та його кордонів. Цей метод можна використовувати лише тоді, коли рішення

сфокусовано на одній конкретній ділянці. Для задачі, поставленої у цій дисертації, таке рішення не є ефективним, адже подібний підхід змусить програмувати кожну камеру окремо, та якщо ми коли-небудь повернемо камеру або захочемо виявити місця для паркування на іншій вулиці, нам доведеться знову вручну закодувати місця для паркування. Тому було обрано автоматичний спосіб виявлення паркувальних місць.

Місце для паркування – це місце, де автомобіль перебуває тривалий час. Тож, оптимальним та універсальним варіантом є виявлення місць, де машини перебувають тривалий час у нерухомому стані.

Якщо ми можемо виявити автомобілі та з'ясувати, які з них не переміщуються кадрами відео, ми можемо змінити розташування паркувальних місць.

Таким чином метод визначення стає універсальним для паркувальних просторів та вулиць з дозволом паркування авто.

За допомогою введення класу потенційних паркувальних місць та постійної перевірки існуючих, алгоритм (рисунок 3.2) також є зручним для оновлення стану доріг, що є важливим для сьогодення.

Перше, що приходить на думку – визначати місце знаходження автомобілів за допомогою Mask R-CNN, бо ця архітектура на сьогоднішній день демонструє найкращі результати у вирішенні задач розпізнавання об'єктів з відеопотоку. Він досить швидкий. Завдяки сучасному графічному процесору ми маємо можливість виявляти об'єкти у відео з високою роздільною здатністю з кількома кадрами в секунду.

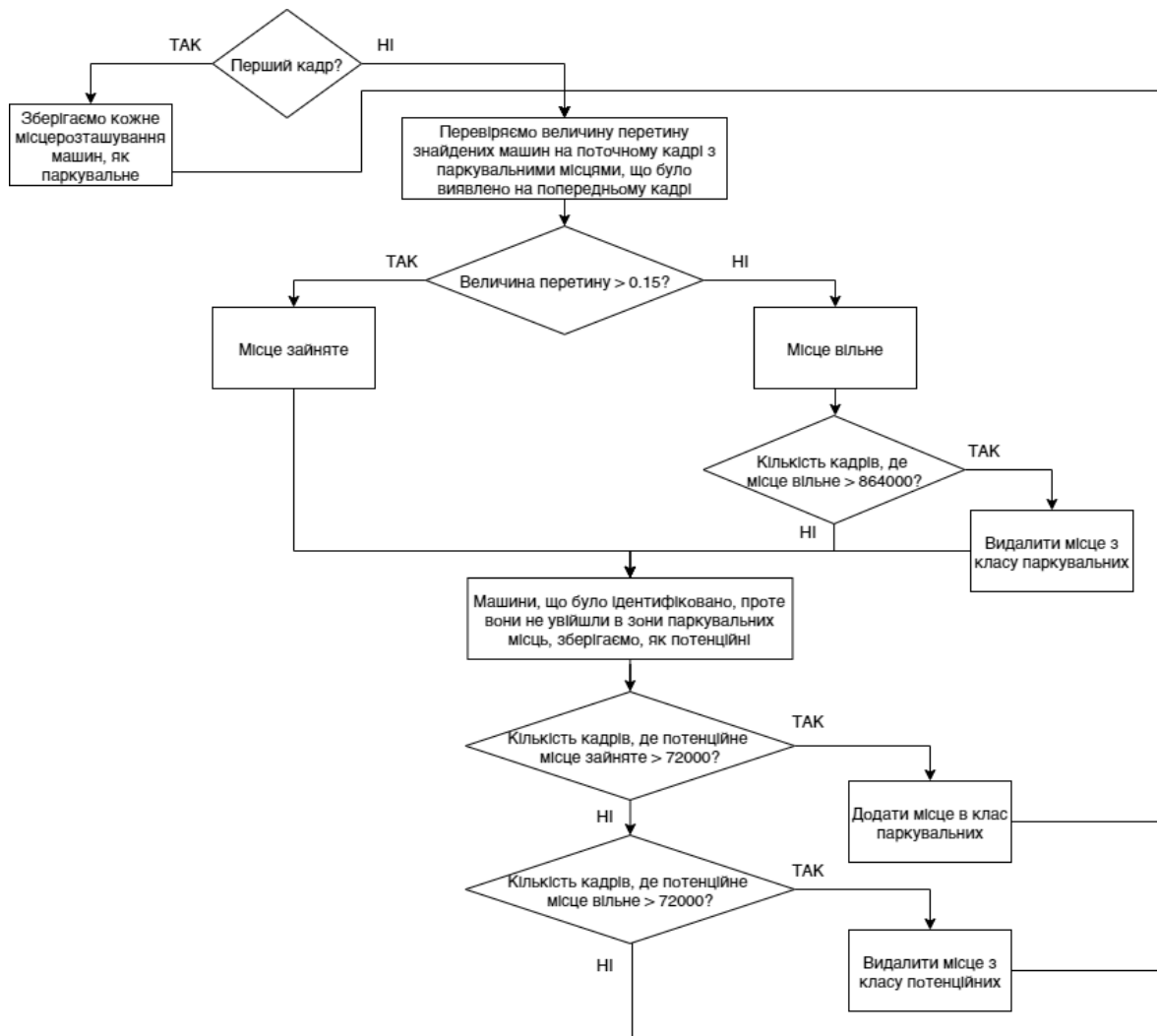


Рисунок 3.2– Алгоритм визначення паркувальних місць

Проте, обмежувальні рамки автомобілів на нашому зображенні частково перекриваються. Тож, якщо ми припустимо, що кожен із цих обмежувальних прямокутників являє собою місце для паркування, можливо, що маска може бути частково зайнята автомобілем, навіть коли місце порожнє.

Також слід врахувати, що наша задача полягає не тільки в тому, щоб виявити автомобіль, а й у фіксуванні паркувальних місць. Тож маючи зафіксовану камеру, аналізувати кожний кадр для пошуку паркувальних місць – не має сенсу. Достатньо визначити положення паркувальних місць, та надалі декілька кадрів аналізувати лише потрібні місця на зображенні. Це дозволить підвищити швидкість обробки.

3.2 Визначення зайнятості паркувального місця

Існує багато підходів машинного навчання, які ми можемо використовувати для виявлення об'єкта на зображенні. Деякі з них було описано у другому розділі:

- HOG (гістограма орієнтованих градієнтів);
- CNN (згортова нейронна мережа);
- Mask R-CNN, Faster R-CNN або YOLO (готові моделі).

Оскільки HOG дуже чуттєвий до змін орієнтації зображення, а готові моделі мають недоліки на невеликих зображеннях, було створено альтернативну архітектуру на основі CNN.

Порівнюючи існуючі архітектури в таблиці 3.1 можна зробити висновок, що наразі найбільш ефективною архітектурою з існуючих для поставленої задачі є AlexNet, адже вона є найбільш швидкою, що важливо для обробки в реальному часі та потребує небагато пам'яті (рисунок 3.3).

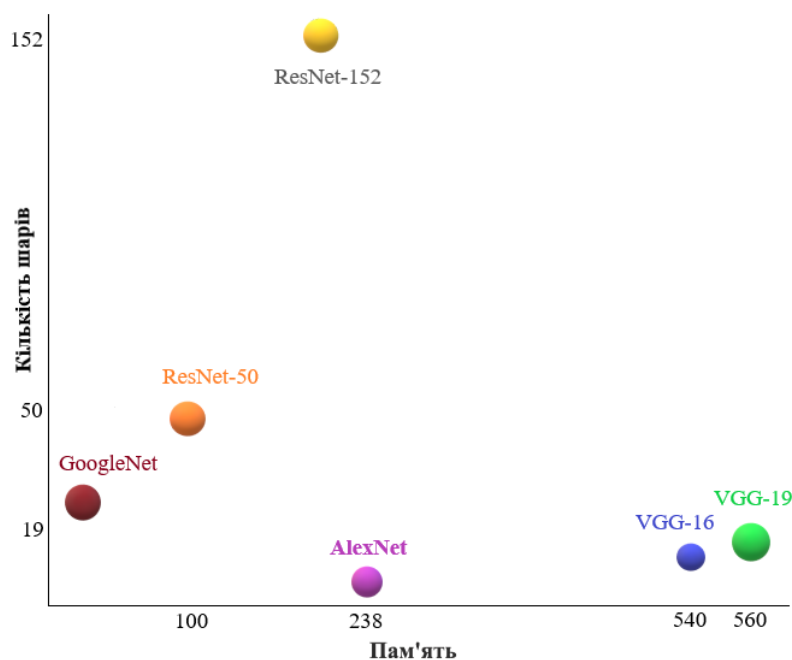


Рисунок 3.3– Порівняння архітектур CNN

Таблиця 3.1 – Порівняння архітектур CNN

Архітектура	Пам'ять (М)	Кількість шарів	Модель
AlexNet	238	8	5 згорткових + 3 повнозв'язних
VGG-16	540	16	13 згорткових + 3 повнозв'язних
VGG-19	560	19	16 згорткових + 3 повнозв'язних
GoogleNet	40	22	21 згорткових + 1 повнозв'язний
ResNet-50	100	50	49 згорткових + 1 повнозв'язний
ResNet-152	235	152	151 згорткових + 1 повнозв'язний

AlexNet є дуже популярною глибокою згортковою нейронною мережею, яка використовується як еталон у багатьох роботах.

Але на практиці використання такої архітектури безпосередньо на пристрої з низьким рівнем обчислювальної потужності є дуже складним завданням, особливо з огляду на той факт, що одній камері може знадобитися відстежувати кілька паркувальних місць одночасно і що кожне паркувальне місце потребує однієї незалежної задачі виявлення зайнятості.

Для того, щоб програмне забезпечення могло працювати ефективніше безпосередньо на камерах, було визначено менш глибинну архітектуру CNN. Таке спрощення мережі також виправдовується тим, що оригінальна архітектура AlexNet (рисунок 3.4) була розроблена для завдань візуального розпізнавання, набагато складніших, ніж наша проблема бінарної класифікації. Спочатку AlexNet навчався на базі даних з мільйона зображень,

бо був створений для набору даних ImageNet, який складався з 22 000 категорій. У нашому випадку ми просто повинні розрізняти два класи.

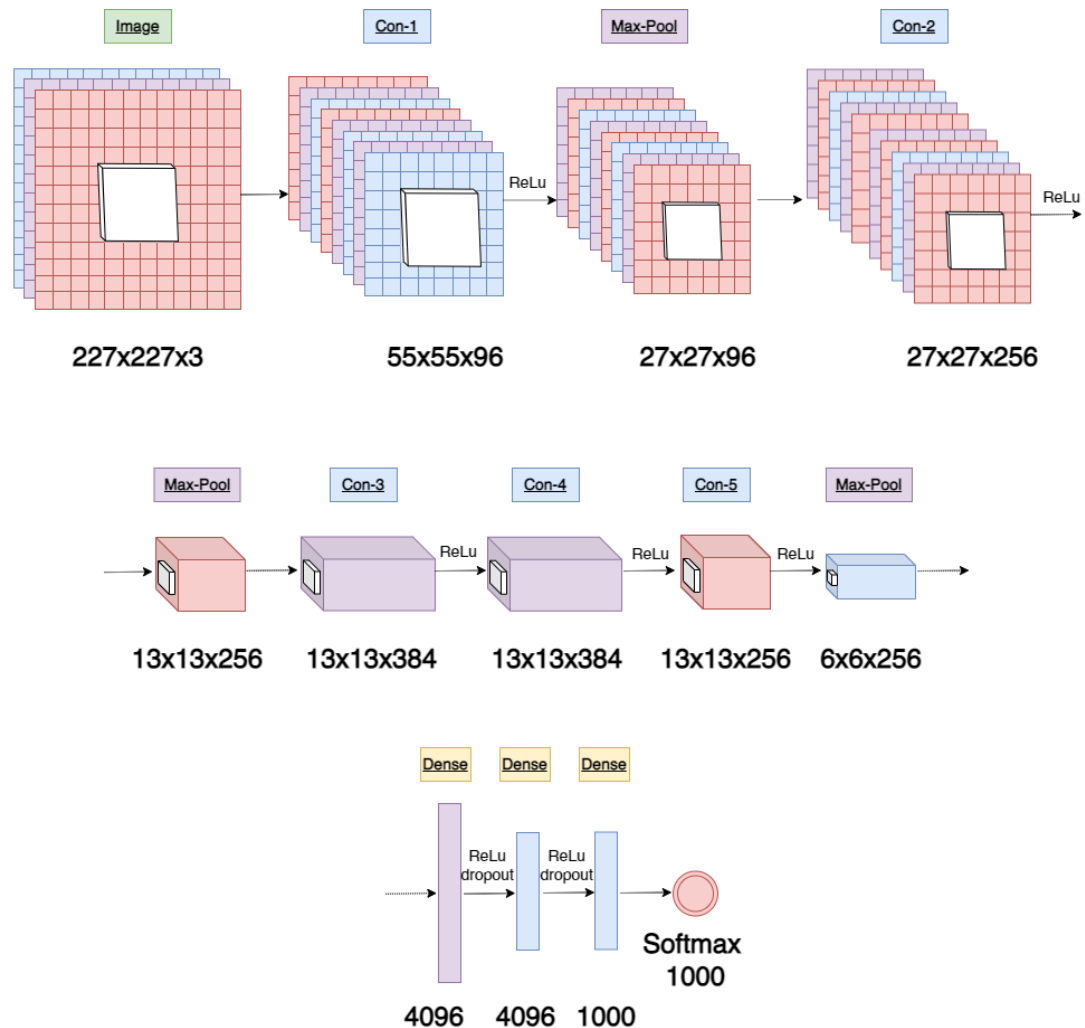


Рисунок 3.4— Архітектура стандартного AlexNet

Отже, у модифікованій архітектурі (рисунок 3.5) було використано три згорткові шари і два повнозв'язані шари, включаючи вихідний шар.

За першим і другим згортковими шарами (con1-2) слідує MaxPool, локальна нормалізація (LRN) і зрізаний лінійний вузол (ReLU). Третій згортковий рівень (con3) не використовує LRN. Кількість шарів і кількість нейронів у шарі різко зменшено, тож було отримано архітектуру з приблизно 22,412,904 параметрами, на відміну від 62,378,344 у AlexNet.

У середньому визначення зайнятості 50 паркувальних місць займає близько 15 секунд.

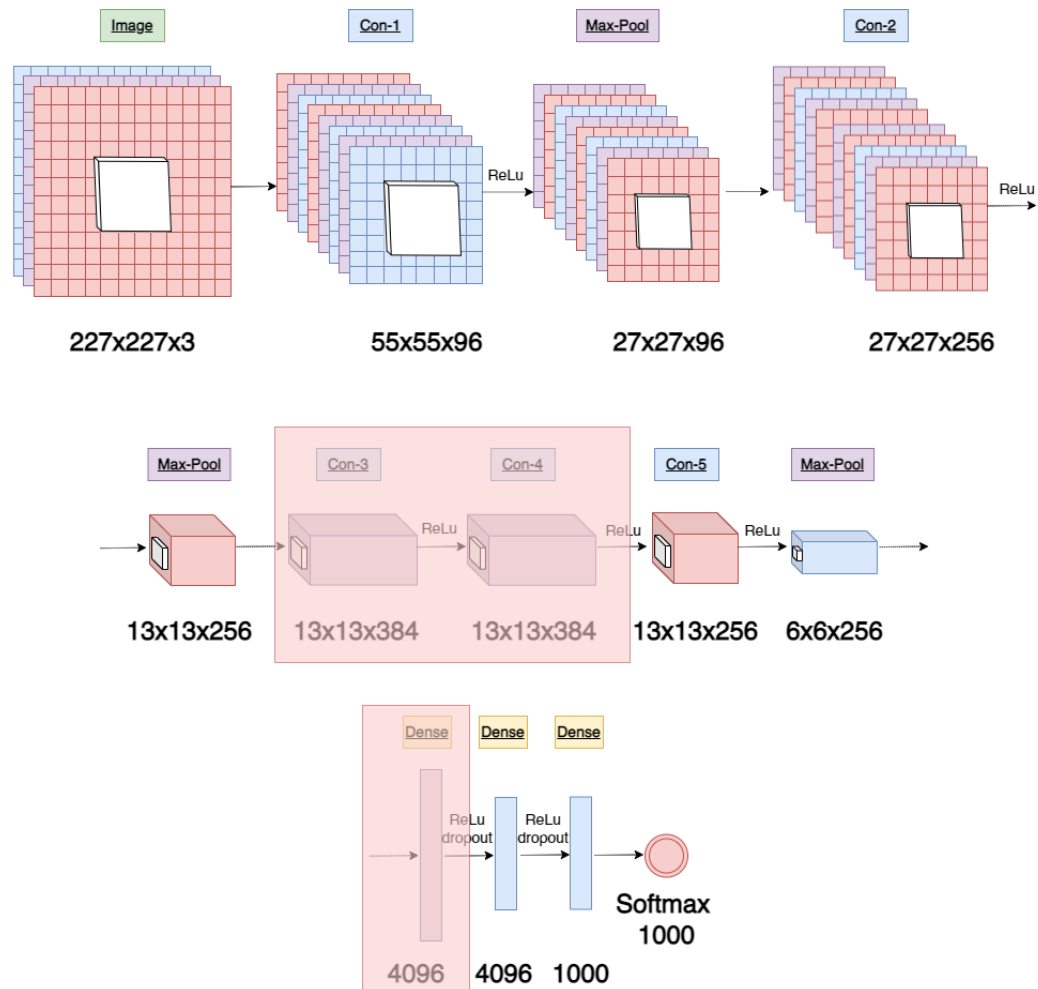


Рисунок 3.5– Модифікована архітектура AlexNet

Отже, завдяки модифікації AlexNet вдалося пришвидшити роботу.

Спробуємо інший підхід (рисунок 3.6): залишимо 3 повнозв'язні шари, 1 вхідний рівень, 1 шар згортки, 1 ReLu, 1 MaxPool. Це допоможе ще скоротити час, необхідний для визначення у реальному часі, та не втратити якість через попередній поділ зображення.

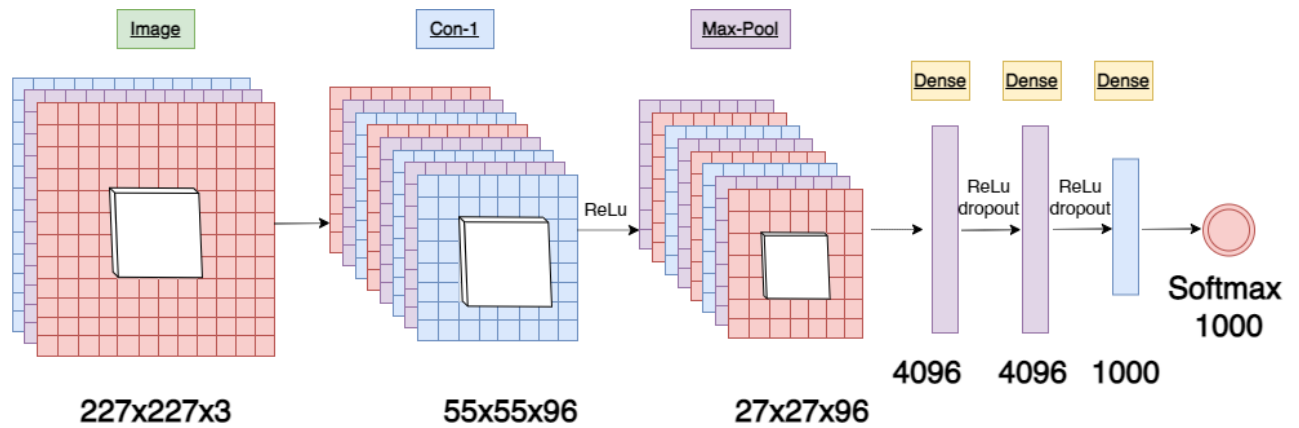


Рисунок 3.6— Модифікована архітектура AlexNet - 2

Наша спеціально розроблена модель складається з одного шару згортки, який виконує операцію згортки над вхідними зображеннями. Вагові показники в шарах згортки навчаються та оновлюються на кожній ітерації навчання за допомогою алгоритму зворотного поширення. Він має крок 4 із ядром 11x11, що дає розмір виходу 55x55. Далі - шар ReLu з сигмоїдною функцією активації, щоб внести нелінійність у модель, як показано в (3)

$$f'(x) = \frac{d}{dx} \ln(1 + e^x) = \frac{1}{1 + e^x}, \quad (3)$$

Після шару ReLu - шар MaxPool, який зменшує дискретизацію. Зменшення розміру призводить до втрати інформації. Однак така втрата є вигідною для мережі з двох причин:

- 1) Зменшення розміру призводить до менших витрат на обчислення для майбутніх шарів мережі та протидіє переобладнанню;
- 2) У кінці ми маємо шар, який з'єднує вихід попереднього шару і дає вихід 2 класу.

У середньому визначення зайнятості 50 паркувальних місць займає близько 8 секунд.

3.3 Запропонована архітектура

Модель складається з двох етапів:

1. Mask R-CNN.
2. Модифікована архітектура AlexNet – 2.

На першому етапі всі зображення відеопотоку обробляються за допомогою Mask R-CNN. Це зроблено для того, щоб отримати потрібні значення координат місцезнаходження попередньо визначених паркувальних місць.

Отримавши значення координат паркувальних місць, вони зберігаються, та на другому етапі передаються до модифікованої архітектури AlexNet – 2, яка визначає зайнятість паркувальних місць, спираючись тільки на зображення вже обмежених конкретних місць.

Періодично зображення проходить Mask R-CNN для того, щоб підтвердити данні місцезнаходження паркувальних місць або додати нові та відкинуті неактуальні місця.

Для навчання Mask R-CNN було використано стандартний набір даних COCO (Common Objects In Context), який містить зображення, які вже анотовані масками об'єктів. У цьому наборі даних є понад 12 000 зображень зі вже окресленими автомобілями.

Для навчання модифікована архітектури було використано набір даних CNRPark + EXT, який було описано раніше.

Розділення CNRPark + EXT на навчальну та перевірочну вибірку можна побачити у таблиці 3.2. Поділ COCO було реалізовано у співвідношенні 20\80.

Остаточний алгоритм зображено на рисунку 3.7.

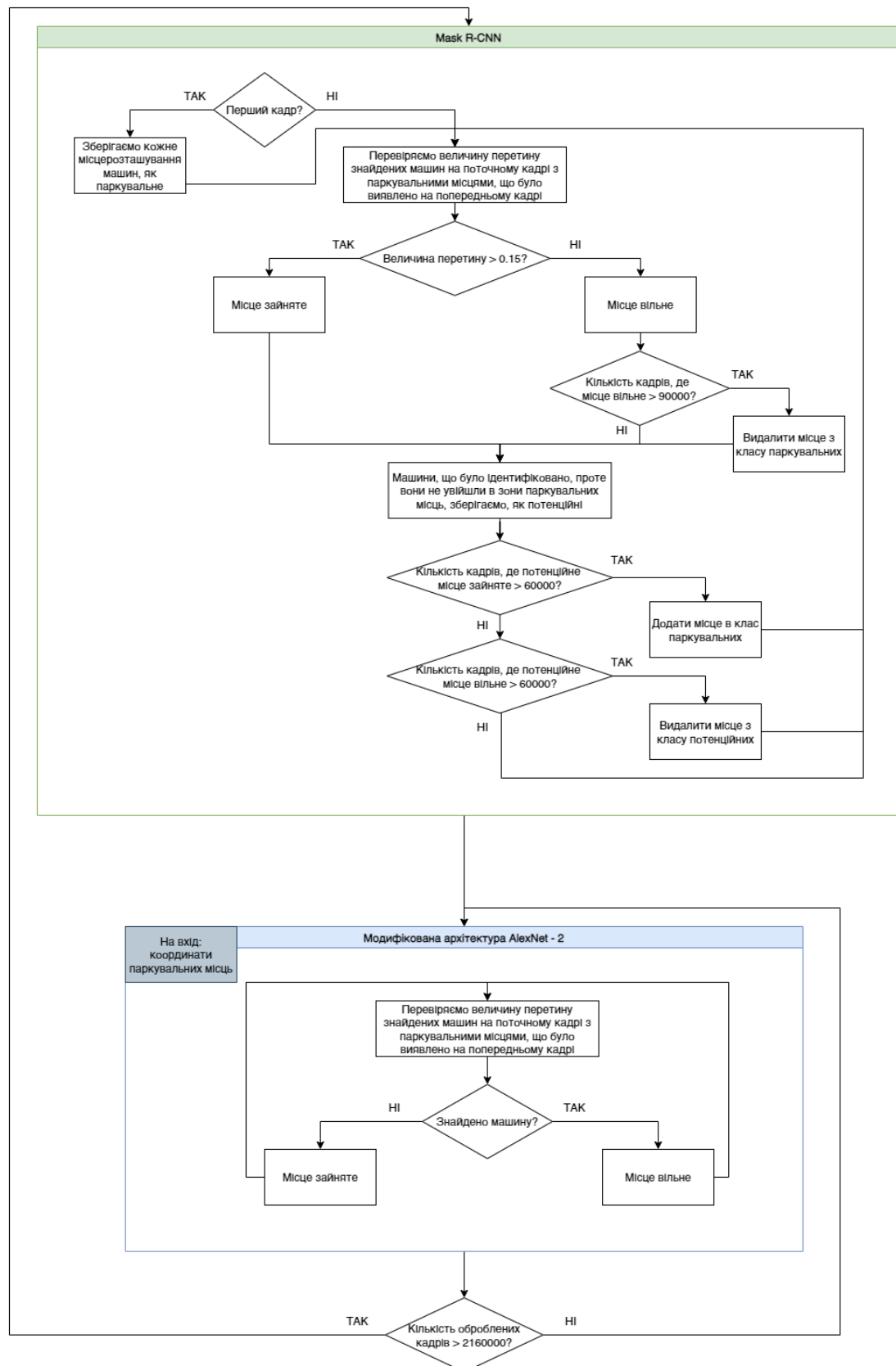


Рисунок 3.7– Запропонований алгоритм виявлення вільних паркувальних місць на основі модифікованої архітектури AlexNet – 2

Таблиця 3.2 –Розподіл на навчальну та перевірочну вибірку CNRPark + EXT

Набір даних	Вільні місця	Зайняті місця	Всього
Навчальна	52,526	63,446	115,972
Перевірочна	13,132	15,861	28,993

3.4 Результати роботи програми

Після першого етапу на зображенні (рисунок 3.8) визначено:

- Обмежувальну рамку об'єкта на зображенні, задану як розташування пікселів X/Y;
- Оцінку впевненості виявлення об'єкта. Чим більше число, тим більше впевненість моделі в тому, що вона правильно ідентифікувала об'єкт.

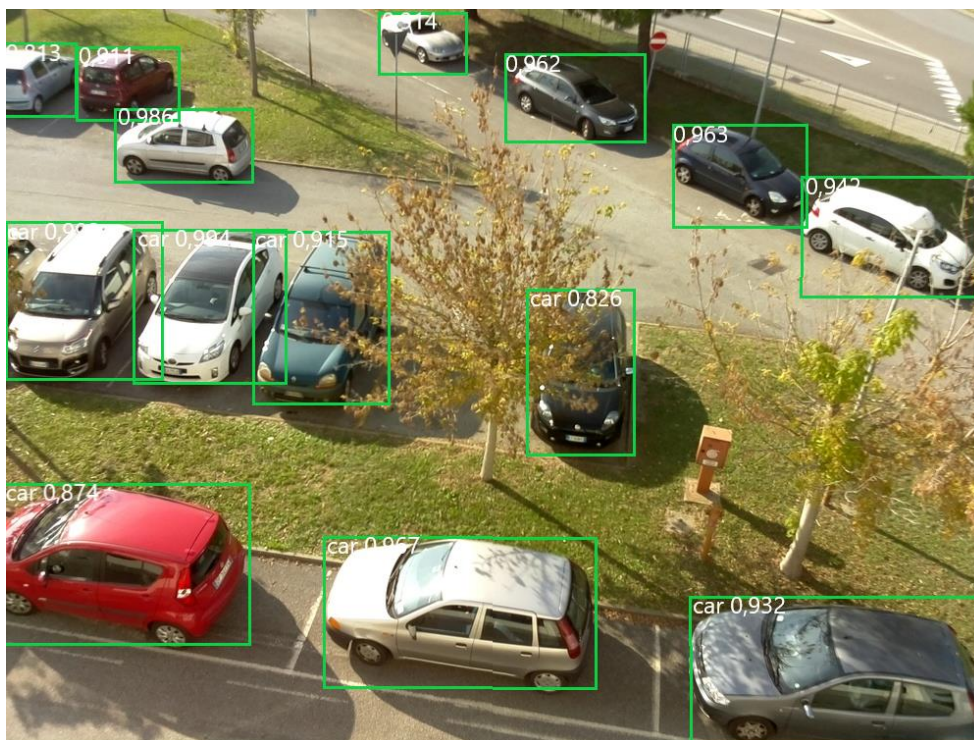


Рисунок 3.8– Визначення паркувальних місць

Після другого етапу на зображенні (рисунок 3.9) збережено ті самі поля, та визначено їх статус.

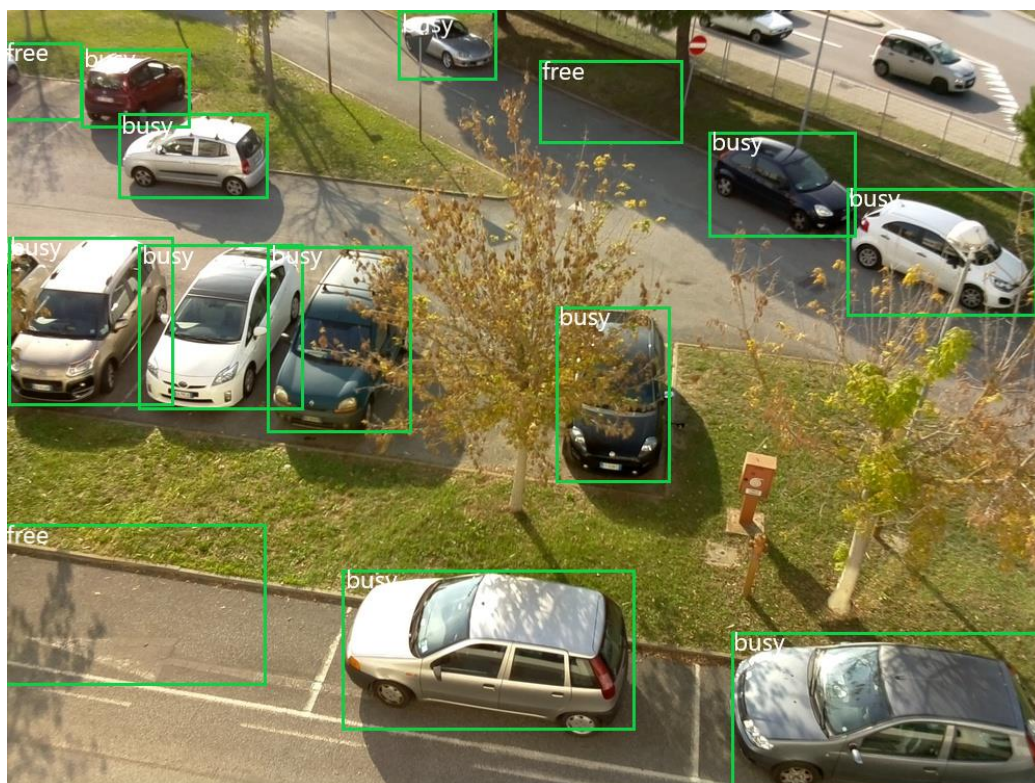


Рисунок 3.9– Визначення статусу паркувальних місць

3.5 Порівняння ефективності запропонованої архітектури з існуючою

Для порівняння ефективності роботи запропонованої моделі було обрано наступні методи та параметри:

- матриця невідповідності визначення статусу паркувального місця (таблиці 3.3-3.4);
- час для визначення статусу паркувального місця (таблиця 3.6);
- точність визначення статусу паркувального місця (таблиця 3.5).

Та наступні моделі:

- Mask R-CNN;
- Mask R-CNN + AlexNet;

- Mask R-CNN + модифікована архітектура.

Таблиця 3.3 –Матриця невідповідності визначення стану паркувального місця Mask R-CNN + AlexNet

	Вільне	Зайняте
Вільне	13132	0
Зайняте	48	15813

Таблиця 3.4 –Матриця невідповідності визначення стану паркувального місця Mask R-CNN + модифікована архітектура

	Вільне	Зайняте
Вільне	12998	134
Зайняте	337	15524

Точність вираховується на основі даних з таблиць 3.3-3.4 за формулою (4).

$$\text{Точність} = \frac{cpt + cnt}{cpt + cnt + cpf + cpn}, \quad (4)$$

де cpt – вірно визначено, що паркувальне місце вільне,
 cnt – вірно визначено, що паркувальне місце зайняте,
 cpf – не вірно визначено, що паркувальне місце вільне,
 cpn – не вірно визначено, що паркувальне місце зайняте.

Значення для Mask R-CNN були взято з існуючого рішення для даної задачі. [11]

Таблиця 3.5 –Порівняння точності визначення статусу паркувальних місць

Моделі	Точність у відсотках
Mask R-CNN + AlexNet	0,998
Mask R-CNN + модифікована архітектура	0,983
Mask R-CNN	0,987

Таблиця 3.6 –Порівняння витраченого часу для визначення статусу паркувальних місць

Моделі	Час
Mask R-CNN	10,32 мс
Mask R-CNN + AlexNet	19,98 мс
Mask R-CNN + модифікована архітектура	6,11 мс

3.6 Висновки

Було запропоновано модель з модифікованою під задачу пошуку вільного паркувального місця архітектурою нейронної мережі CNN.

Через зменшення кількості згорткових шарів вдалося пришвидшити роботу системи та значно понизити значення обчислювальної потужності процесора, на якому буде працювати програма. Така зміна надасть можливість використання даної системи навіть на відеореєстраторах з невеликою кількістю пам'яті та потужністю.

Нова модифікація дає змогу результативно працювати з камерами, що мають частоту кадрів 17 кадрів в секунду.

Щодо точності: звичайно, зменшенням кількості згорткових шарів ми негативно повпливали на точність, проте цей вплив був незначний, через те, що задачу було зведено до бінарної, для якої повністю вистачає запропонованої архітектури.

Таким чином, в роботі вдалося знайти рішення, ефективніше за існуюче (Mask R-CNN), та підібрати модель, яка буде працювати значно краще на пристроях з гіршими технічними показниками, що значно зменшить потрібні витрати.

РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

4.1 Опис ідеї проєкту

Стартап проєкт, що базується на даній роботі має назву «CamParking».

Даний розділ описує економічну обґрунтованість імплементації цього проєкту. В таблиці приведено опис ідеї стартап-проєкту (таблиця 4.1).

Таблиця 4.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Ідея полягає в створенні системи розпізнавання вільних паркувальних місць	Інтеграція з міськими та приватними системами відеоспостереження	Економія часу на пошук паркувального місця
	Інтеграція з паркувальними системами для приватних паркувальних зон	Зменшення вартості реалізації даного функціоналу в місті та приватних паркувальних зон
	Інтеграція з додатками для оплати паркування в місті	

Аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;

- визначення попереднього кола конкурентів або товарів-замінників, що вже існують на ринку, та проведення збору інформації щодо значень техніко-економічних показників.

Визначений перелік слабких та сильних характеристик та властивостей ідеї потенційного товару є основою для формування його конкурентоспроможності (таблиця 4.2).

Таблиця 4.2 –Визначення сильних, слабких характеристик ідеї проєкту

№ п/п	Техніко- економічні характеристики ідеї	Товари/концепції конкурентів				W (слабка сторона)	S (сильна сторона)
		Власний проєкт	Parkeon	Мікком AS101 ProPark	Smart parking		
1.	Можливість визначити точне місцезнаходження вільного паркувального місця	+	-	+	-	Іноді показники можуть бути не дуже точними	Визначення статусу паркувального місця та його розташування можлива без додаткового технічного обладнання
2.	Можливість беззмінного використання на різних зонах	+	+	-	+	Поки що система не протестована для використання декількома камерами на одній зоні	Не потрібне переналаштування системи для використання на різних зонах

4.2 Технологічний аудит ідеї проєкту

Проведено аудит технології, за допомогою якої реалізовано ідею проєкту.

Визначення технологічної здійсненності проєкту передбачає аналіз таких складових:

- за якою технологією буде виготовлено товар згідно ідеї проєкту;
- чи існують такі технології, чи їх потрібно розробити/доробити;
- чи доступні такі технології авторам проєкту.

Аналіз проведено в таблиці (таблиця 4.3).

Таблиця 4.3 –Технологічна здійсненність ідей проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Використання існуючого технічного обладнання у багатьох зонах, які є основними цільовими замовниками	Відеокамери та відеореєстратор	Наявна	Доступна
2.	Розпізнавання вільних паркувальних місць	Навчання нейронної мережі на спеціалізованих вибірці	Наявна	Доступна

Обрана технологія реалізації ідеї проєкту:

Так, проєкт можна починати реалізовувати. Варто реалізувати відсутню наразі оптимальну архітектуру нейронної мережі, адже саме він є дуже важливим і основним для швидкої та ефективної роботи системи.

4.3 Аналіз ринкових можливостей запуску стартап-проєкту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проєкту, та ринкових загроз, які можуть перешкодити реалізації проєкту, дозволяє спланувати напрями розвитку проєкту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проєктів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

Рентабельність — поняття, що характеризує економічну ефективність виробництва, за якої за рахунок грошової виручки від реалізації продукції (робіт, послуг) повністю відшкодовує витрати на її виробництво й одержується прибуток як го. Значення середньої рентабельності визначено за формулою:

$$R = \frac{P}{I \times n} \times 100\%, \quad (5)$$

де P – прибуток за час експлуатації проєкту,

I – повна сума інвестиційних витрат,

n – час експлуатації проєкту.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців (враховуючи вже існуючі і додатки), од	5-7
2.	Загальний обсяг продаж, грн/ум.од	1000000
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Необхідно вирішити питання з архітектурою, перш ніж виходити на ринок
5.	Специфічні вимоги до стандартизації та сертифікації	Немає
6.	Середня норма рентабельності в галузі (або по ринку), %	25-30%

За результатами аналізу таблиці 4.4 було зроблено висновок, що ринок є відносно привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики, та сформовано орієнтовний перелік вимог до товару для кожної групи (таблиця 4.5).

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проєкту, та факторів, що йому перешкоджають (таблиці 4.6, 4.7).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Водії прагнуть витратити менше часу для пошуку вільного паркувального місця	Водії	Водії прагнуть витратити менше часу та пального на паркування машини	Необхідно забезпечити ефективність роботи системи
2.	Власники паркувальних зон прагнуть витратити менше грошей на вдосконалення та розширення функціональних можливостей	Житлові комплекси, торгівельно-розважальні центри, приватні паркувальні зони, власники великих паркувальних зон	Власники паркувальних зон прагнуть поліпшити умови паркування за невеликі кошти	Необхідно забезпечити простоту інтеграції

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому

впровадженню проєкту, та факторів, що йому перешкоджають (таблиці 4.6, 4.7).

Ринкові можливості - це сприятливі обставини, які підприємство може використовувати для отримання переваг. Як приклад ринкових можливостей, можна привести погіршення позицій конкурентів, різке зростання попиту, появу нових технологій виробництва продукції, зростання рівня доходів населення і т. п. Слід зазначити, що можливостями з погляду SWOT-аналізу є не всі можливості, які існують на ринку, а тільки ті, які можна використовувати.

Ринкові загрози - події, настання яких може несприятливо вплинути на підприємство. Приклади ринкових загроз: вихід на ринок нових конкурентів, зростання податків, зміна смаків покупців, зниження народжуваності і т. п.

Чинники попиту - тут доцільно взяти до уваги місткість ринку, темпи його зростання або скорочення, структуру попиту на продукцію підприємства і т. ін.

Чинники конкуренції - слід врахувати кількість основних конкурентів, наявність на ринку товарів-замінників, висоту бар'єрів входу на ринок і виходу з нього, розподіл ринкових часток між основними учасниками ринку і т.ін.

Чинники збуту - необхідно надати увагу кількості посередників, наявності сіток розподілу, умовам поставок матеріалів та комплектуючих і т. ін.

Економічні чинники - враховується курс гривні (долара, євро), рівень інфляції, зміна рівня доходів населення, податкова політика держави і т.ін.

Політичні і правові чинники - оцінюється рівень політичної стабільності в країні, рівень правової письменності населення, рівень законослухняності, рівень корумпованості влади і т.ін.

Науково-технічні чинники - звичайно береться до уваги рівень розвитку науки, ступінь упровадження інновацій(нових товарів, технологій) в промислове виробництво, рівень державної підтримки розвитку науки і т. ін.

Соціально-демографічні чинники - слід врахувати чисельність і статеву та вікову структури населення регіону, в якому працює підприємство, рівень народжуваності і смертності, рівень зайнятості населення і т.ін.

Таблиця 4.6 – Характеристика потенційних клієнтів стартап-проекту

Параметри оцінки	Можливості	Загрози
Попит	Адаптація до ринку, що розширяється, проведення кампаній по залученню більшої кількості людей для користування системою. Можлива реакція на загрозу: Переорієнтація продукту разом з переорієнтацією ринку	На ринку безліч конкурентів, що мають велику ЦА, рівень фінансування та обладнання, що дозволить їм швидше виходити на ринок
Конкуренція	Оскільки рішення є кращим за декількома важливими показниками, є можливість проаналізувати клієнтів інших компаній та розширити кількість користувачів	Можлива поява конкурентів зі схожим підходом до вирішення та більшим досвідом роботи у сфері
Науково-технічні	Є можливість розвивати та покращувати архітектуру з появою більш інноваційних підходів до вирішення подібних задач	Зі швидким розвитком у сфері є велика ймовірність появи конкурентів на ринку з більш якісним рішенням
Соціально-культурні	Діджиталізація суспільства	Скептичне відношення старшого покоління до якості та надійності нейронних мереж

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 4.7).

Таблиця 4.7 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - досконала	Існує до 7 фірмконкуренток на ринку	Пошук слабких сторін конкурентів, та альтернативної ніші на ринку врахувати ціни конкурентних компаній та, використовуючи рекламу, вказати на конкретні переваги перед конкурентами; Пошук шляхів покращення якості продукту
2. За рівнем конкурентної боротьби - міжнародне	Компанії предствляють різні країни	Використання хмарних послуг для доступності програмного забезпечення в різних куточках світу
3. За галузевою ознакою - внутрішньогалузева	Конкуренти мають рішення, Які використовується лише всередині даної галузі	Розширення функціоналу для вирішення нових задач аналізу.
4. Конкуренція за видами послуг: - товарно-родова	Конкуренція на рівні технології задоволення потреб	Розробити продукт з урахуванням недоліків конкурентів.
5. За характером конкурентних переваг - нецінова - цінова	Користувачу важлива низька ціна на послуги і точність, швидкість розпізнавання	Слід зосередити зусилля на розробку власної системи з викоистанням безплатних бібліотек, платформ та дешевого обладнання
6. За інтенсивністю - не марочна	Продається втілення ідеї, а не певний бренд	Просування ідеї програми на державному рівні та серед власників приватних паркувальних зон

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (таблиця 4.8). М. Портер вирізняє п'ять основних факторів (Рисунок 4.1), що впливають на привабливість вибору ринку з огляду на характер конкуренції.

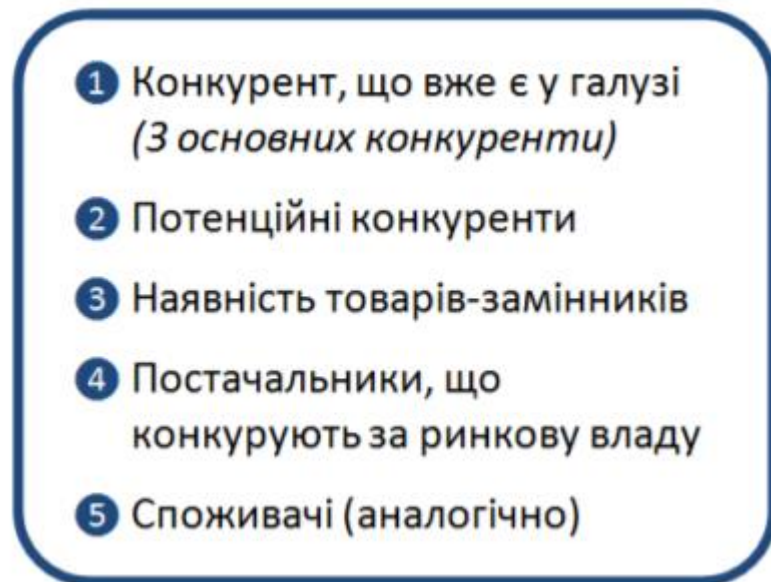


Рисунок 4.1– Модель Портера

Сильні позиції компанії за кожним з факторів означають її можливості забезпечити необхідні темпи обороту капіталу та її здатність впливати на інших агентів ринку, диктуючі їм власні умови співпраці. Характеристики факторів моделі відрізняються для різних галузей та змінюються із часом. Сила кожного фактору є функцією від структури галузі та її техніко-економічних характеристик.

Економічні чинники - враховується курс гривні (долара, євро), рівень інфляції, зміна рівня доходів населення, податкова політика держави і т.ін.

Політичні і правові чинники - оцінюється рівень політичної стабільності в країні, рівень правової письменності населення, рівень законослухняності, рівень корумпованості влади і т.ін.

Таблиця 4.8 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Parkeon, Мікком AS101 ProPark, Smart parking	Інші системи зі схожими функціями	Можлива інтеграція з системами відеоспостереження	Кількість зацікавлених клієнтів	Поява схожих дешевих і якісних продуктів-конкурентів
Висновки:	Є, але потенційно наш проєкт матиме в якості і ціні.	Можливості входу в ринок присутні, необхідно вирішити проблему вибору архітектури, потенційні конкуренти – можуть використати наші знання та помилки для створення кращого продукту		Важливим для користувача є точність і швидкість роботи системи	Коли з'являться конкуренти чи товари-замінники, модифікована архітектура для прискорення роботи програми стане ключовим фактором успіху проєкту.

За результатами аналізу таблиці 4.8 було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Цей висновок був врахований при формулюванні переліку факторів конкурентноспроможності у наступному пункті.

На основі аналізу конкуренції, проведеного в таблиці 4.8, а також із урахуванням характеристик ідеї проєкту (таблиця 4.2), вимог споживачів до товару (таблиця 4.5) та факторів маркетингового середовища (таблиці 4.6, 4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності.

Аналіз оформлюється за таблицею (таблиці 4.9-4.10).

Таблиця 4.9 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1.	Фактор часу	Окремі частини концепції є новими, для перейняття ідеї та втілення її у життя, потенційним конкурентам знадобиться час
2.	Фактор ціни	Чим менша ціна послуги тим більший попит а отже і вага на ринку.

Таблиця 4.10 – Порівняльний аналіз сильних та слабких сторін «CamParking»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з CamParking						
			-3	-2	-1	0	+1	+2	+3
1.	Фактор часу			+					
2.	Фактор ціни			+					

Фінальним етапом ринкового аналізу можливостей впровадження проєкту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та

слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 4.11).

Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 4.11 – SWOT- аналіз стартап-проєкту

Сильні сторони: Якість послуг Новизна підходу Необхідні витрати для реалізації та використання	Слабкі сторони: Іноді результати можуть бути недостатньо точними
Можливості: Створення лідера серед конкурентних пропозицій	Загрози: Посилення конкуренції на ринку

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проєкту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проєкти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 4.12).

Таблиця 4.12 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Ціль: отримання прибутку в короткостроковій перспективі Конкуренція: цінова Взаємодія з фірмами: активна боротьба за долю ринку, що належить конкурентам	В короткостроковому плані – велика В довгостроковому плані – значний ризик втратити долю ринку, якщо займатися лише ціновою конкуренцією. В основному треба зосередитись на якості	Рік
2.	Ціль: захоплення частини ринку, підтримання її розміру та поступове нарощення об'ємів Конкуренція: нецінова Взаємодія з конкурентами: співпраця, активний моніторинг їх діяльності, при можливій появі реальних конкурентів можна запропонувати злиття компаній/проектів	Висока імовірність отримання ресурсів та утримання їх протягом довгого проміжку часу. Більш імовірний розвиток компанії та постійне покращення продукту.	Рік

Обрано альтернативу 2 як ту, що забезпечує довгострокове життя проєкту.

4.4 Розроблення ринкової стратегії проєкту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 4.13).

Таблиця 4.13 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції всегменті	Простота входу у сегмент
1.	Районні адміністрації міста	Користувачі готові споживати продукт, адже це недорогий крок до діджиталізації району	Потенційно високий	Невелика конкуренція, адже на даний момент схожих проєктів не розглядається	Досить просто
2.	Власники паркувальних зон	Користувачі готові споживати продукт і потребують його для зменшення витрат	Високий попит	Велика конкуренція, але мій проєкт має більш розширений функціонал	За наявності реклами – досить просто

За результатами аналізу потенційних груп споживачів було обрано цільову групи, для якої буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію концентрованого маркетингу.

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку.

За М. Портером, існують три базові стратегії розвитку, що відрізняються за ступенем охоплення цільового ринку та типом конкурентної переваги, що має бути реалізована на ринку (за витратами або визначними якостями товару).

Стратегія лідерства по витратах передбачає, що компанія за рахунок чинників внутрішнього і/або зовнішнього середовища може забезпечити більшу, ніж у конкурентів маржу між собівартістю товару і середньо-ринковою ціною (або ж ціною головного конкурента). Зокрема, ця стратегія припускає, що за рахунок великих можливостей по об'ємах збуту товарів (портфеля укладених контрактів на постачання) і продуктивності підприємство може добитися менших витрат. Ця стратегія зазвичай тісно пов'язана з можливістю досягнення ефекту масштабу і досвіду.

Компанії, що вибирають цю стратегію, проводять ретельний контроль за постійними витратами, знижують виробничі, збутові і рекламні витрати, проводять інвестиції, спрямовані на зменшення витрат, ретельне опрацювання конструкцій нових товарів.

Переваги стратегії за Ж.-Ж. Ламбеном:

- фірма здатна протистояти своїм прямим конкурентам навіть у разі цінової війни і в змозі отримувати прибуток при ціні, мінімально допустимій для конкурентів;
- сильні клієнти не можуть добитися зниження ціни нижче рівня, прийняттого для найбільш сильного конкурента;
- низькі витрати забезпечують захист проти сильних постачальників, оскільки дають фірмі велику гнучкість у разі підвищення вхідних витрат;

- низькі витрати створюють бар'єр входу для нових конкурентів і одночасно хороший захист проти товарів-замінників. В ході конкурентної боротьби з використанням цієї стратегії з ринку вимушені будуть піти фірми, менш ефективні з точки зору величини і структури витрат, нездібні до проведення технологічних новацій, спрямованих на зниження витрат. Стратегія диференціації передбачає надання товару важливих з точки зору споживача відмінних властивостей, які роблять товар відмінним від товарів конкурентів. Така відмінність може базуватися на об'єктивних або суб'єктивних, відчутних і невідчутних властивостях товару(у ширшому розумінні – комплексі маркетингу), бути реальною або уявною. Інструментом реалізації стратегії диференціації є ринкове позиціонування.

Переваги стратегії за Ж.-Ж. Ламбенем:

- По відношенню до прямих конкурентів диференціація знижує ступінь замінності товару, посилює прихильність марці, зменшує чутливість до ціни і тим самим підвищує рентабельність;
- Прихильність клієнтів послабляє їх тиск на фірму і перешкоджає приходу на ринок нових конкурентів;
- Підвищена рентабельність збільшує стійкість до можливого зростання витрат в результаті дійсильного постачальника;
- Відмінні властивості товару і завойована прихильність клієнтів захищають фірму і від товарів-замінників. Реалізація цієї стратегії вимагає, як правило, більш високих витрат. Проте успішна диференціація дозволяє компанії домогтись більшої рентабельності за рахунок того, що ринок готовий прийняти більш високу ціну (цінову премію бренду). При веденні конкурентної боротьби з використанням цієї стратегії на ринку в першу чергу терплять фіаско фірми, що не здатні визначати потреби цільових ринків, оперативно реагувати на зміни в ринковому попиті, проводити ефективну політику маркетингових комунікацій, не мають необхідних навичок в області брендингу.

Стратегія спеціалізації передбачає концентрацію на потребах одного цільового сегменту, без прагнення охопити увесь ринок. Мета тут полягає в задоволенні потреб вибраного цільового сегменту краще, ніж конкуренти. Така стратегія може спиратися як на диференціацію, так і на лідерство по витратах, або і на те, і на інше, але тільки у рамках цільового сегменту. Проте низька ринкова доля у разі невдалої реалізації стратегії може істотно підірвати конкурентоспроможність компанії.

Визначення стратегії розвитку приведено у таблицях (таблицях 4.14-4.15).

Таблиця 4.14 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
2.	Захоплення частини ринку, підтримання її розміру та поступове нарощення об'ємів	Стратегія концентрованого маркетингу	Новизна послуг Доступність продукту Простота в користуванні продуктом	Диференціація

Таблиця 4.15 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
2.	Ні	Обидва підходи	Так, а саме – визначення наявності вільних місць	Стратегія лідера

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. таблицю 4.5), а також в залежності від обраної базової стратегії розвитку (таблиця 4.15) та стратегії конкурентної поведінки (таблиця 4.16) розроблено стратегію позиціонування (таблиця 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Позиціонування — це маркетингове забезпечення товарів бажаного місця на ринку і у свідомості потенційних покупців (образ). Позиція компанії чи продукту показує чим він унікальний УТП (унікальну торговельну пропозицію), чим відрізняється від конкурентів (відстройка від конкурентів), чим корисний споживачу.

З точки зору маркетингу товар являє собою сукупність відчутних (розмір, колір, маса, швидкість і т. п.) і невловимих (престижність, модність, сучасність і т. п.) властивостей.

Стратегія спеціалізації передбачає концентрацію на потребах одного цільового сегменту, без прагнення охопити увесь ринок. Мета тут полягає в задоволенні потреб вибраного цільового сегменту краще, ніж конкуренти. Така стратегія може спиратися як на диференціацію, так і на лідерство по витратах, або і на те, і на інше, але тільки у рамках цільового сегменту. Проте низька ринкова доля у разі невдалої реалізації стратегії може істотно підірвати конкурентоспроможність компанії. Найважливішими здібностями, які повинна мати компанія, що приймає цю стратегію, є з генерування маркетингових ідей, здійснення продуктових новацій.

Дослідження свідчать, якщо позиціонування здійснюється більше ніж за трьома ознаками, то воно є неефективним, оскільки не відкладається у свідомості споживача.

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартапкомпанії на ринку.

Вибір стратегії позиціонування приведено в таблиці (таблиця 4.16).

Таблиця 4.16 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту
1.	Функціонування програми в будь-який час, без помилок	Диференціація	Залучення великої бази даних, для навчання, що зробить виявлення швидшим і точнішим; створення власної архітектури та моделі для швидкої роботи	Швидка робота; якісна робота

Продовження таблиці 4.16

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
2.	Наявність необхідних пристроїв		Проект потребує лише камери та відеореєстратор, які є в наявності в більшості паркувальних зон	Дешева інтеграція

4.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у таблиці 4.17 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.17 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1.	Швидке, якісне та доступне виявлення	Нова швидка в роботі архітектура та велика вибірка для навчання	Висока точність виявлення, простота інтеграції

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 4.19).

1-й рівень - при формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язане з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень - цей рівень являє рішення того, як буде реалізований товар в реальному/включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень - товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні.

Опис трьох рівнів зображено в таблиці (таблиця 4.18).

Таблиця 4.18 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система, що дозволяє швидко та точно знайти вільні паркувальні місця не залежно від місцевості		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1.функція вибору кращої моделі		
	2. функція вибору кращої архітектури		
	Якість: за рахунок комбінування моделей		
	Марка: CamParking		
III. Товар із підкріпленням	До продажу		
	Після продажу		
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок закритості вихідного коду додатку, захист ідеї товару.			

Після формування маркетингової моделі товару слід відмітити, що проєкт буде захищено від копіювання за допомогою ноу-хау. Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проєкту), яке передбачає аналіз ціни на товари-аналоги або товари субституту, а також аналіз рівня доходів цільової групи споживачів (таблиця 4.19). Аналіз проведено експертним методом.

Таблиця 4.19 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	200-1000 грн/день	-	2000-4000грн/день	-

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 4.20): проводити збут власними силами і залучати сторонніх посередників, користуватися однорівневим каналом збуту.

Таблиця 4.20 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Купівля та плата за підтримку	Поширення продукту, інформування	Однорівневий	Залучена

Стратегія лідерства по витратах передбачає, що компанія за рахунок чинників внутрішнього і/або зовнішнього середовища може забезпечити більшу, ніж у конкурентів маржу між собівартістю товару і середньоринковою ціною (або ж ціною головного конкурента). Зокрема, ця стратегія припускає, що за рахунок великих можливостей по об'ємах збуту товарів (портфеля укладених контрактів на постачання) і продуктивності підприємство може добитися менших витрат. Ця стратегія зазвичай тісно пов'язана з можливістю досягнення ефекту масштабу і досвіду.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 4.21).

Таблиця 4.21 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Перед вирішенням задачі виявлення вільних паркувальних місць, клієнти шукають в Інтернеті корисні поради щодо сервісів, які можуть їм знадобитися, і дотримуються рекомендацій.	Соціальні мережі (facebook, twitter, linkedin); тендери	Точність та швидкість виявлення, легкість у інтеграції	Впевнити клієнта у тому, що CamParking – найкращий спосіб організації паркувальної зони	Повідомлення у соціальних мережах, короткі ролики

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проєкт, та відповідну обрану альтернативу ринкової поведінки.

4.6 Висновки

У даному розділі було проведено аналіз програмного продукту у якості стартап-проєкту. Можна зазначити що у проєкта є можливість до комерціалізації, адже він може зацікавити, як великі компанії, які мають власні паркувальні зони, так і, наприклад, бажання водіїв одного дому пришвидшити пошук вільного паркувального місця біля дому. Він є універсальним до технічного обладнання та місцевості.

На ринку наявна конкуренція, існує декілька фірмконкурентів, тому вихід на нього не буде легким. Але проєкт є конкурентоспроможним завдяки розширенню можливостей, своїй вартості та швидкості роботи.

Для впровадження ринкової реалізації проєкту слід обрати альтернативу, яка передбачає розробку програмного продукту, а потім якісну рекламу та PR, сконцентровану навколо позитивних характеристик даного програмного продукту, таких як низька ціна, надійність, безпека і т.д.

З огляду на проведений аналіз, можна чітко сказати, що подальша імплементація проєкту є доцільною, адже він знайде свою цільову аудиторію та займе місце на ринку, та буде прибутковим.

ВИСНОВКИ

Завдяки співпраці штучного інтелекту з автомобільною промисловістю були розроблені різні алгоритми для покращення роботи систем розпізнавання вільних місць на паркувальному просторі.

Вже створено та працюють на практиці декілька готових рішень з різними підходами: від різних датчиків до комп'ютерного зору.

Проте не було знайдено жодної системи, яка б задовольняла поставлену задачу: пошуку або створення бюджетної системи, що знаходить вільні паркувальні місця.

Оскільки в Україні на даний момент часу все більше поширюється встановлення відеокамер на дорогах та паркувальних місцях, було прийнято рішення розробити систему на основі комп'ютерного зору, чийм головним і єдиним для використання інструментом, будуть відеокамери.

Однак існує певна неоднозначність у виборі найефективнішого алгоритму проєктування таких систем через варіації експериментальних платформ та відсутність загальних наборів даних.

Тож було створено програмний продукт на базі авторської модифікованої архітектури CNN та порівняно результати роботи з системою на базі найкращої стандартної архітектури для вирішення цієї проблеми на сьогоднішній день AlexNet за параметрами продуктивності: точність виявлення вільних місць, швидкість обробки алгоритму.

Побудований сервіс може бути використаний як на вулицях, так і для паркувальних площадок торговельних центрів або житлових комплексів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Deng W., Luo X., Jiang L., Luo Y. Research on video-based monitoring algorithm of parking spaces. *Third International Conference on Multimedia Information Networking and Security*, November, 2011. P. 261–264.
2. Wolff J., Heuer T., Gao H., Weinmann M., Voit S., Hartmann U. Parking monitor system based on magnetic field sensors. *IEEE Intelligent Transportation Systems Conference*. 2006. P. 1275–1279.
3. Krizhevsky A., Sutskever I., Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 2012. Vol. 25, No. 1. P. 1097–1105.
4. Chen S., Wang H., Xu F., Jin Y.Q. Target classification using the deep convolutional networks for SAR images. *IEEE Transactions on Geoscience and Remote Sensing*. 2016. Vol. 54, No. 9. P. 4806–4817.
5. Zhou Y., Wang H., Xu F., Jin Y.Q. Polarimetric SAR image classification using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*. 2016. Vol. 13, No. 12. P. 1935–1939.
6. Kim B.K., Kang H.S., Park S.O. Drone classification using convolutional neural networks with merged Doppler images. *IEEE Geoscience and Remote Sensing Letters*. 2017. Vol. 14, No. 1. P. 38–42.
7. Acharya D., Yan W., Khoshelham K. Real-time image-based parking occupancy detection using deep learning. *Proc. of the 5th Annual Research@Locate Conf.*, Adelaide, Australia, 2018. P. 33–40.
8. Kubler S., Robert J., Hefnawy A., Cherifi C., Bouras A. IoT-based smart parking system for sporting event management. *Proc. of the 13th Int. Conf. on Mobile and Ubiquitous Systems, Computing, Networking and Services*, Hiroshima, Japan, 2016. P. 104–114.
9. Karami E., Prasad S., Shehata M. Image Matching Using SIFT, SURF, BRIEF, and ORB: Performance Comparison for Distorted Images. *Proceedings of the 2015 Newfoundland Electrical and Computer Engineering*

- Conf.*. URL: <http://pzs.dstu.dp.ua/ComputerGraphics/d&d/bibl/Karami.pdf> (дата звернення: 14.10.2021).
10. Ma Y., Gu G., Yin B., Qi S., Chen K. Support vector machines for the identification of real-time driving distraction using in-vehicle information systems. *Journal of Transportation Safety & Security*. 2020. Vol. 121, No. 12. P. 1–24.
 11. Iqbal K., Abbas S., Khan M. A., Ather A., Khan M. S. Autonomous parking-lots detection with multi-sensor data fusion using machine deep learning techniques. *Computers, Materials & Continua*. 2021. Vol. 66, No. 2. P. 1595–1612.
 12. Marso, K., Macko, D. A New Parking-Space Detection System Using Prototyping Devices and Bluetooth Low Energy Communication. *Int. J. Eng. Technol. Innov.*. 2019. Vol. 9, No. 2. P. 108.
 13. Ho G.T.S., Tsang Y.P., Wu C.H., Wong W.H., Choy K.L. A computer vision-based roadside occupation surveillance system for intelligent transport in smart cities. *Sensors*. URL: <https://www.mdpi.com/1424-8220/19/8/1796> (дата звернення: 01.11.2021).
 14. Mumtaz Z., Ullah S., Ilyas Z., Aslam N., Iqbal S., Liu S., Meo J., Madni H. An automation system for controlling streetlights and monitoring objects using Arduino. *Sensors*. URL: <https://doi.org/10.3390/s18103178> (дата звернення: 05.11.2021).

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```

import os

import numpy as np

import cv2

import mrcnn.config

import mrcnn.utils

from mrcnn.model import MaskRCNN

from pathlib import Path

from twilio.rest import Client


# Configuration that will be used by the Mask-RCNN library
class MaskRCNNConfig(mrcnn.config.Config):
    NAME = "coco_pretrained_model_config"

    IMAGES_PER_GPU = 1

    GPU_COUNT = 1

    NUM_CLASSES = 1 + 80 # COCO dataset has 80 classes + one background

class

    DETECTION_MIN_CONFIDENCE = 0.6


# Filter a list of Mask R-CNN detection results to get only the detected cars / trucks

```

```

def get_car_boxes(boxes, class_ids):
    car_boxes = []

    for i, box in enumerate(boxes):
        # If the detected object isn't a car / truck, skip it
        if class_ids[i] in [3, 8, 6]:
            car_boxes.append(box)

    return np.array(car_boxes)

# Twilio config
twilio_account_sid = 'YOUR_TWILIO_SID'
twilio_auth_token = 'YOUR_TWILIO_AUTH_TOKEN'
twilio_phone_number = 'YOUR_TWILIO_SOURCE_PHONE_NUMBER'
destination_phone_number = 'THE_PHONE_NUMBER_TO_TEXT'
client = Client(twilio_account_sid, twilio_auth_token)

# Root directory of the project
ROOT_DIR = Path(".")

# Directory to save logs and trained model

```

```

MODEL_DIR = os.path.join(ROOT_DIR, "logs")

# Local path to trained weights file
COCO_MODEL_PATH = os.path.join(ROOT_DIR, "mask_rcnn_coco.h5")

# Download COCO trained weights from Releases if needed
if not os.path.exists(COCO_MODEL_PATH):
    mrcnn.utils.download_trained_weights(COCO_MODEL_PATH)

# Directory of images to run detection on
IMAGE_DIR = os.path.join(ROOT_DIR, "images")

# Video file or camera to process - set this to 0 to use your webcam instead of a
video file
VIDEO_SOURCE = "test_images/parking.mp4"

# Create a Mask-RCNN model in inference mode
model = MaskRCNN(mode="inference", model_dir=MODEL_DIR,
config=MaskRCNNConfig())

# Load pre-trained model
model.load_weights(COCO_MODEL_PATH, by_name=True)

```

```
# Location of parking spaces
parked_car_boxes = None

# Load the video file we want to run detection on
video_capture = cv2.VideoCapture(VIDEO_SOURCE)

# How many frames of video we've seen in a row with a parking space open
free_space_frames = 0

# Have we sent an SMS alert yet?
sms_sent = False

# Loop over each frame of video
while video_capture.isOpened():
    success, frame = video_capture.read()
    if not success:
        break

    # Convert the image from BGR color (which OpenCV uses) to RGB color
    rgb_image = frame[:, :, ::-1]

    # Run the image through the Mask R-CNN model to get results.
    results = model.detect([rgb_image], verbose=0)
```

```
# Mask R-CNN assumes we are running detection on multiple images.
# We only passed in one image to detect, so only grab the first result.
r = results[0]
```

```
# The r variable will now have the results of detection:
# - r['rois'] are the bounding box of each detected object
# - r['class_ids'] are the class id (type) of each detected object
# - r['scores'] are the confidence scores for each detection
# - r['masks'] are the object masks for each detected object (which gives you the
object outline)
```

```
if parked_car_boxes is None:
    # This is the first frame of video - assume all the cars detected are in parking
spaces.

    # Save the location of each car as a parking space box and go to the next frame
of video.

    parked_car_boxes = get_car_boxes(r['rois'], r['class_ids'])
else:
    # We already know where the parking spaces are. Check if any are currently
unoccupied.
```

```
# Get where cars are currently located in the frame
car_boxes = get_car_boxes(r['rois'], r['class_ids'])
```

```
# See how much those cars overlap with the known parking spaces
```

```

overlaps = mrcnn.utils.compute_overlaps(parked_car_boxes, car_boxes)

# Assume no spaces are free until we find one that is free
free_space = False

# Loop through each known parking space box
for parking_area, overlap_areas in zip(parked_car_boxes, overlaps):

    # For this parking space, find the max amount it was covered by any
    # car that was detected in our image (doesn't really matter which car)
    max_IoU_overlap = np.max(overlap_areas)

    # Get the top-left and bottom-right coordinates of the parking area
    y1, x1, y2, x2 = parking_area

    # Check if the parking space is occupied by seeing if any car overlaps
    # it by more than 0.15 using IoU
    if max_IoU_overlap < 0.15:
        # Parking space not occupied! Draw a green box around it
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 3)
        # Flag that we have seen at least one open space
        free_space = True
    else:
        # Parking space is still occupied - draw a red box around it
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 1)

```

```

# Write the IoU measurement inside the box
font = cv2.FONT_HERSHEY_DUPLEX
cv2.putText(frame, f"{max_IoU_overlap:0.2}", (x1 + 6, y2 - 6), font, 0.3,
(255, 255, 255))

```

```

# If at least one space was free, start counting frames
# This is so we don't alert based on one frame of a spot being open.
# This helps prevent the script triggered on one bad detection.
if free_space:
    free_space_frames += 1
else:
    # If no spots are free, reset the count
    free_space_frames = 0

# If a space has been free for several frames, we are pretty sure it is really
free!

```

```

if free_space_frames > 10:
    # Write SPACE AVAILABLE!! at the top of the screen
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(frame, f"SPACE AVAILABLE!", (10, 150), font, 3.0, (0, 255,
0), 2, cv2.FILLED)

```

```

# If we haven't sent an SMS yet, sent it!
if not sms_sent:
    print("SENDING SMS!!!")

```

```
message = client.messages.create(  
    body="Parking space open - go go go!",  
    from_=twilio_phone_number,  
    to=destination_phone_number  
)  
sms_sent = True
```

```
# Show the frame of video on the screen  
cv2.imshow('Video', frame)
```

```
# Hit 'q' to quit  
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

```
# Clean up everything when finished  
video_capture.release()  
cv2.destroyAllWindows()
```