

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ЛУЧКА Святослав Ігорович

**Веб-застосунок дистанційного інтелектуального виявлення
вільного паркомісця / Web application for remote intelligent
detection of free parking spaces**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-41
С.І. Лучка

Науковий керівник:
к.т.н., доцент Х.В. Лип'яніна-
Гончаренко

Дипломний проект допущено до захисту
«__» _____ 2023 р.

Завідувач кафедри
_____ М.П. Комар

Тернопіль – 2023

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
М.П. Комар
« _____ » _____ 2022р.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ
Лучці Святославу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема проекту: Веб-застосунок дистанційного інтелектуального виявлення вільного паркомісця / Web application for remote intelligent detection of free parking spaces

керівник проекту к.т.н., доцент Х.В. Лип'яніна-Гончаренко

затверджені наказом по університету від 08 грудня 2022 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

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

- визначення проблеми пошуку паркомісць у великих містах
- аналіз існуючих рішень для виявлення вільних парковок
- провести моделювання функціональної та структурної складових веб-застосунку
- розробити алгоритм дистанційного виявлення вільних паркомісць
- розробити підключення до відео потоку
- провести навчання моделі на основі глибокого навчання з підкріпленням методом градієнтного спуску
- розробити веб-інтерфейс застосунку для дистанційного інтелектуального виявлення вільного паркомісця.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

– алгоритм дистанційного виявлення вільних паркомісць

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	к.т.н., доцент Х.В. Лип'яніна-Гончаренко		

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області і постановка задачі дослідження	30.12.2022	
2	Розробка функціоналу та алгоритмів веб-застосунку	24.03.2023	
3	Програмно-технологічне забезпечення	12.05.2023	
4	Повне завершення та оформлення дипломного проекту	01.06.2023	

Студент _____ С.І. Лучка
(підпис)

Керівник проекту _____ Х.В. Лип'яніна-Гончаренко
(підпис)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 69 с., 27 рис., 2 додатки, 50 джерел.

Метою дипломного проекту є розробка веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць у великих містах з метою підвищення ефективності паркування та зменшення трафіку на дорогах.

Об'єктом дослідження є процес паркування в містах-мільйонниках, а предметом дослідження є розробка веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць.

Предметом дослідження є розробка веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць у великих містах.

Розроблено та досліджено програмне забезпечення для покращення ефективності паркування та зменшенні трафіку на дорогах у великих містах-мільйонниках, що призведе до зменшення заторів та покращення якості життя мешканців міст.

Також ця робота може мати значення для компаній, що займаються послугами доставки та таксі, а також для міських влад, що прагнуть до покращення умов паркування та розв'язання проблем трафіку на дорогах.

**ДИСТАНЦІЙНЕ ВИЯВЛЕННЯ ПАРКОМІСЦЬ, ВЕБ-ЗАСТОСУНОК,
МАШИННЕ НАВЧАННЯ, РОЗПІЗНАВАННЯ ОБРАЗІВ.**

ABSTRACT

The bachelor's thesis report: 69 pages, 27 figures, 2 appendices, 50 references.

The aim of the diploma project is to develop a web application for remote intelligent detection of available parking spaces in large cities, with the goal of improving parking efficiency and reducing traffic congestion.

The object of research is the parking process in million-cities, while the subject of research is the development of a web application for remote intelligent detection of available parking spaces.

The subject of research is the development of a web application for remote intelligent detection of available parking spaces in large cities.

Software has been developed and investigated to improve parking efficiency and reduce traffic congestion in large million-cities, which will lead to reduced congestion and improved quality of life for city residents.

This work may also be relevant to companies involved in delivery services and taxi services, as well as municipal authorities seeking to improve parking conditions and address traffic problems.

REMOTE PARKING DETECTION, WEB APPLICATION, MACHINE LEARNING, IMAGE RECOGNITION.

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Веб-застосунок дистанційного інтелектуального виявлення вільного паркомісця.

1.2 Область застосування – покращення паркування та зменшення трафіку у великих містах.

2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою дипломного проекту є розробка веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць у великих містах з метою підвищення ефективності паркування та зменшення трафіку на дорогах.

4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмної системи:

– вибір даних (пакетних даних / даних у реальному часі) з різних архітектур великих даних;

– розробка підключення до відео потоку

- навчання моделі на основі глибокого навчання з підкріпленням методом градієнтного спуску
- розробка веб-інтерфейс застосунку для дистанційного інтелектуального виявлення вільного паркомісця

5.2 Вимоги до апаратних засобів:

- кластер запускається на фізичній машині Macbook Pro з 16 гігабайт оперативної пам'яті та процесором 2.3 GHz Intel Core i5.

5.3 Вимоги до програмних засобів:

- для розробки програмне забезпечення - Python 3.7;
- для створення графічного інтерфейсу користувача використано – tkinter, Adobe XD;
- для реалізації моделей навчання – фреймворк sklearn.

6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання _____ С.І. Лучка
(підпис) (прізвище та ініціали)

Керівник дипломного проекту _____ Х.В. Ліп'яніна-Гончаренко
(підпис) (прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
Вступ.....	11
1 Аналіз предметної області і постановка задачі дослідження.....	13
1.1 Визначення проблеми пошуку паркомісць у великих містах	13
1.2 Огляд існуючих рішень для виявлення вільних парковок	17
1.3 Вибір перспективного шляху і постановка задачі дослідження	22
2 Розробка функціоналу та алгоритмів веб-застосунку.....	24
2.1 Моделювання функціональної та структурної складових веб-застосунку	24
2.2 Алгоритм дистанційного виявлення вільних паркомісць	27
2.3 Метод класифікації індивідуальних паркувальних місць	29
3 Програмно-технологічне забезпечення	34
3.1 Підключення до відео потоку	34
3.2 Навчання моделі на основі глибокого навчання з підкріпленням методом градієнтного спуску	39
3.3 Веб-інтерфейс застосунку для дистанційного інтелектуального виявлення вільного паркомісця	52
Висновки	62
Список використаних джерел.....	64
Додаток А.....	70
Додаток Б	71

					<i>ДП.КН.8091455.069.ПЗ</i>									
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>										
<i>Розроб.</i>		<i>Лучка С.</i>			Веб-застосунок дистанційного інтелектуального виявлення вільного паркомісця									
<i>Перевір.</i>		<i>Ліп'яніна-Гончаренко</i>												
<i>Консульт.</i>														
<i>Н. Контр.</i>		<i>Ліп'яніна-Гончаренко</i>												
<i>Затверд.</i>		<i>Комар М.П.</i>												
					<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><i>Літ.</i></td> <td style="text-align: center;"><i>Аркуш</i></td> <td style="text-align: center;"><i>Аркушів</i></td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">8</td> <td style="text-align: center;">69</td> </tr> <tr> <td colspan="3" style="text-align: center;"><i>ЗУНУ.ФКІТ.КН-41</i></td> </tr> </table>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>		8	69	<i>ЗУНУ.ФКІТ.КН-41</i>		
<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>												
	8	69												
<i>ЗУНУ.ФКІТ.КН-41</i>														

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- IoT: Internet of Things - Інтернет речей, концепція, що передбачає підключення фізичних пристроїв (речей) до Інтернету та обмін даними між ними.
- MILP: Mixed Integer Linear Programming - цілочисельне лінійне програмування, метод оптимізації, який використовується для розв'язання задач з великою кількістю обмежень та цілочисельних змінних.
- DFD: Data Flow Diagram - діаграма потоку даних, графічне зображення, що ілюструє потік даних та обробку інформації у системі.
- БД: База даних - організована колекція даних, яка зберігається та управляється в електронному вигляді.
- GPS: Global Positioning System - система глобального позиціонування, що використовує супутники для визначення географічного положення точки на Землі.
- GD: Gradient Descent - градієнтний спуск, алгоритм оптимізації, що використовується для знаходження мінімуму (або максимуму) функції шляхом крокового руху в напрямку найшвидшого спаду (або найшвидшого зростання) градієнту.
- ML: Machine Learning - машинне навчання, галузь штучного інтелекту, що вивчає розробку алгоритмів та моделей, що дозволяють комп'ютерам самостійно вчитися та покращувати свою продуктивність на основі даних.
- DL: Deep Learning - глибоке навчання, підгалузь машинного навчання, що використовує нейронні мережі з багатьма шарами для автоматичного витягування високорівневих ознак з даних.
- API: Application Programming Interface - інтерфейс програмування додатків, набір специфікацій та інструментів, що дозволяє різним програмним додаткам взаємодіяти один з одним.

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

- URL: Uniform Resource Locator - єдиноадресний локатор ресурсу, стандартизований спосіб вказати адресу веб-ресурсу або файлу в Інтернеті.
- HTTP-запит: Hypertext Transfer Protocol Request - запит, який клієнтська програма (браузер або інше програмне забезпечення) робить до сервера для отримання ресурсів або виконання операцій.
- GET: метод HTTP-запиту, який використовується для отримання (читання) ресурсу з сервера.
- POST: метод HTTP-запиту, який використовується для відправлення (запису або оновлення) даних на сервер.
- ASP.NET Core: фреймворк розробки веб-додатків, що базується на платформі .NET Core.
- OpenCVSharp.dnn: бібліотека комп'ютерного зору, яка надає функціональність глибокого навчання для розпізнавання образів та обробки зображень.

					<i>ДП.КН.8091455.069.ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>10</i>

ВСТУП

Актуальність теми. Проблема пошуку вільних парковочних місць у містах є дуже актуальною. Це стає особливо гострою високому трафіку, який зростає з кожним роком. Пошук вільних парковочних місць може забирати значний час і додатково ускладнювати проблему заторів на дорогах. Розробка веб-застосунку, який зможе дистанційно виявляти вільні парковочні місця, може стати ефективним рішенням проблеми.

Використання інтелектуальних методів аналізу даних, таких як машинне навчання та обробка зображень, дозволяє створити надійний інструмент для виявлення вільних парковочних місць. Веб-застосунок на основі таких методів може бути ефективним інструментом для водіїв, що допоможе значно скоротити час, витрачений на пошук парковки, та зменшити навантаження на дорожні мережі.

Отже, розробка веб-застосунку дистанційного інтелектуального виявлення вільного паркомісця є дуже актуальною і має великий потенціал для вирішення проблем транспортної інфраструктури міст.

Метою дипломного проекту є розробка веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць у великих містах з метою підвищення ефективності паркування та зменшення трафіку на дорогах.

Досягнення цієї мети зумовило потребу теоретичних розробок, визначення та послідовного вирішення таких завдань:

1. визначення проблеми пошуку паркомісць у великих містах
2. аналіз існуючих рішень для виявлення вільних парковок
3. провести моделювання функціональної та структурної складових веб-застосунку
4. розробити алгоритм дистанційного виявлення вільних паркомісць
5. розробити підключення до відео потоку
6. провести навчання моделі на основі глибокого навчання з підкріпленням методом градієнтного спуску

									ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата						11

7. розробити веб-інтерфейс застосунку для дистанційного інтелектуального виявлення вільного паркомісця

Об'єктом дослідження є процес паркування в містах-мільйонниках, а предметом дослідження є розробка веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць.

Предметом дослідження є розробка веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць у великих містах.

Методи дослідження: аналіз існуючих рішень та технологій в галузі паркування; проведення опитування серед користувачів та водіїв, що регулярно користуються послугами паркування; розробка та тестування алгоритмів для виявлення вільних паркомісць за допомогою машинного навчання; розробка веб-застосунку для відображення вільних паркомісць та роботи з користувачами..

Практичне значення полягає в покращенні ефективності паркування та зменшенні трафіку на дорогах у великих містах-мільйонниках, що призведе до зменшення заторів та покращення якості життя мешканців міст. Також ця робота може мати значення для компаній, що займаються послугами доставки та таксі, а також для міських влад, що прагнуть до покращення умов паркування та розв'язання проблем трафіку на дорогах.

Структура та обсяг роботи. Дипломного проекту складається із вступу, трьох розділів, висновків, списку використаних джерел. Повний обсяг роботи становить 69 ст. комп'ютерного тексту, який включає 27 рисунки. Список використаних джерел із 50 найменувань викладено на 6 сторінках.

					ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Визначення проблеми пошуку паркомісць у великих містах

Більшість приватних транспортних засобів проводять більшу частину часу в стані спокою, або в робочий час, або вночі. Це означає, що для кожного автомобіля в місті має бути два місця, де буде припарковано. Ці два місця мають бути в обох кінцях кожної поїздки. Проблеми паркування в містах і міських районах стають все більш важливими і були однією з найбільш обговорюваних тем як широкою громадськістю, так і професіоналами. Дисбаланс між пропозицією паркування та попитом на паркування вважається основною причиною проблем з паркуванням у мегаполісах. Крім того, система паркування відіграє ключову роль у столичній системі дорожнього руху, а її відсутність свідчить про закритий зв'язок із заторами, дорожньо-транспортними пригодами та забрудненням навколишнього середовища. Хоча ефективна система паркування може покращити міський транспорт та міське середовище, а також підвищити якість життя громадян, проблема паркування є аспектом міського планування та транспорту, який часто не береться до уваги. Містобудівники повинні шукати більш ефективні та інноваційні рішення проблеми паркування на рівні управління, планування та проектування.

Проблема паркування в містах та містах фактично означає, що існує розрив між попитом на паркування (кількість автомобілів, які потребують паркувальних місць) та пропозицією паркування (кількість паркувальних місць, достатня для автомобілів, які потребують паркування). Цей розрив пояснюється кількома причинами:

- Більшість старих та історичних міст, особливо столиці, були сплановані вузькими вуличками, де не було машин, а були вози, які рухалися кінями. Крім того, щільність населення цих міст у той час була не такою

					ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

високою порівняно з нинішньою щільністю тих самих міст та тих самих районів. Оскільки міські вулиці не можуть бути змінені або змінені з часом, за винятком деяких важливих причин і в обмежених випадках, ці вузькі вулиці стають відповідальними за розміщення всіх видів транспортних засобів з високою щільністю для руху та паркування, навантаження, яке перевищує їх заплановані потужності.

- Концентрація діяльності та об'єктів, які вимагають високої швидкості автомобілів в одній і тій же зоні, наприклад, концентрація комерційних об'єктів з офісними будівлями та урядовими установами в міських або районних центрах. (Центр Каїра є прикладом цього).

- У нових містах і нових запланованих міських районах завжди спостерігається прорахунок попиту на паркування, очікуваного в цих районах через несподівано підвищений рівень володіння автомобілями, особливо серед населення високого і середнього класів достатку. Це пов'язано з нездатністю системи громадського транспорту, запропонованої в цих нових районах, змусити людей цих класів залежати від неї в своїх робочих або туристичних поїздках.

- Також у планах нових міст та нових міських територій тенденція надання паркувальних місць завжди залежить від паркування на узбіччі та паркувальних майданчиків на рівні вулиць. Не виділено земельних ділянок під паркувальні споруди з декількома поверхами, щоб поглинути зростаючу кількість автомобілів, які шукають парковку, принаймні в районах з зосередженими об'єктами. Наприклад, в місті Новий Каїр і вздовж Дев'яностий авеню немає єдиної землі, виділеної під паркувальну споруду, незважаючи на численні офісні будівлі і торгові центри, розтягнуті на кілька кілометрів.

- У старих та існуючих містах, а також внаслідок вторгнення та явища сукцесії, зміна використання з використання з низьким рівнем автомобілів, таких як житлові, на використання з високим рівнем автомобілів, таких як комерційні чи ділові, також сприяють проблемі. (Замалек і Мохандессін в Каїрі є прикладами цього).

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

- Порушення будівельних норм і правил зонування - які передбачають для кожної місцевості конкретне використання будівель і конкретну поверховість з наданням гаражів в підвалах. Це порушення сприяє зміні всіх розрахунків, встановлених проектувальниками для забезпечення достатньої кількості паркувальних місць для автомобілів на цих ділянках.

Проблема паркування спотворює міський дизайн. Збільшення кількості позавуличних паркувальних майданчиків поблизу або біля кожної будівлі віддаляє будівлі один від одного та збільшує відстані між будівлями, роблячи водіння більш необхідним та ускладнюючи ходьбу. Також естетично безперервні приємні вуличні фасади стають практично неможливими через кілька зазорів цих фасадів між будівлями, створюваних паркувальними зонами [1]. Паркування на узбіччі, яке відбувається рядами перед будівлями, спотворює та змінює шлях пішоходів до цих будівель через їхні входи, як це зроблено. Через відсутність достатніх площ для паркування відкриті майданчики, такі як громадські площі, громадські поля, місця громадських зібрань, з часом і під тиском проблеми перетворюються на паркувальні зони. Паркінги, як правило, вважаються найменш гламурним і найбільш шкідливим для навколишнього середовища видом землекористування. Як рішення для паркування, наземні ділянки часто руйнують відчуття огорожі, необхідного для деяких приміщень та площ у місті, і дозволяють цим огорожам втратити чіткість. З міркувань вартості парковки також рідко реалізуються з таким рівнем деталізації, який підходить для громадської площі. З цієї причини краще відокремлювати ділянки поверхні за будівлями або екранувати парканами, стінами або живоплотами, щоб замаскувати наявність цих ділянок. Паркування на узбіччі робить кілька смуг дороги, зайнятих автомобілями і тим самим збільшує навантаження на інші дорожні смуги, які іноді стають однією єдиною смугою, що призводить до накопичення автомобілів понад проектну пропускну здатність доріг і викликає затори. Крім того, щоб забезпечити більше смуг для паркування на узбіччі, не порушуючи потік транспорту, особливо на вузьких вулицях або в центрі міста, місцева влада, як правило,

					ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

створює ці смуги за рахунок ширини тротуарів, що робить їх нестандартними та не мають деяких функцій. Використання цих тротуарів пішоходами стає складним, незручним і неприємним.

У багатьох містах водії в напружений час не можуть знайти місце для паркування своїх автомобілів у певних районах, таких як КБР та бізнес-центри, або на парковці, або на узбіччі. У цьому випадку розчаровані водії кружляють довколишні квартали в пошуках місця для паркування своїх автомобілів. Це явище називається крейсерським. Круїз має пов'язаний вплив на проблему паркування. Люди, які курсують вулицями в певному районі, щоб знайти місця для паркування своїх автомобілів, вже прибули, але їхні автомобілі під час круїзів роблять вулиці переповненими, споживають більше палива, викидають додаткове забруднення та накладають додаткові затримки на інших людей, які використовують ті самі вулиці для простого переміщення. Слід зазначити, що, з іншого боку, надмірні парковки опосередковано впливають на природне середовище. Стверджується, що темне покриття автостоянок сприяє погіршенню якості води, дратує вплив теплового острова, підвищує температуру повітря та споживає землю. Автостоянки, як правило, вважаються найменш гламурним і найбільш шкідливим для навколишнього середовища видом землекористування [2].

Парковка для автомобілів може бути передбачена в трьох видах; паркінг на вулиці (бордюрний паркінг), позавуличний паркінг (автостоянка) та паркувальні конструкції, які можуть бути двох типів; або як одно- або двоповерхові поверхи, що складають підвал будівлі (якщо підземний) або подіум (якщо надземний), або як індивідуальну споруду (багатоповерховий гараж). Кожна форма паркування має кілька типів в обігу та облаштуванні автомобілів, щоб максимізувати кількість автомобілів, які можуть бути припарковані в ньому. Кожна система паркування в будь-якій з трьох попередніх форм має три ключові частини; кількість, якість та управління. Звичайне планування паркування, як правило, зосереджується в першу чергу на кількості. Він передбачає, що більше завжди краще, і занадто багато ніколи

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

не може бути. Цей тип планування покладається в першу чергу на мінімальні вимоги до паркування та забезпечення рясного паркування. Планування паркування також враховує якість паркувальних зон, таку як зручність та безпека ходьби від паркувального місця до місця призначення, привабливість та безпеку паркувальних об'єктів, а також екологічні вимоги паркувальних зон. Управління паркуванням спрямоване на досягнення найкращих економічних показників паркувальних місць, особливо на стоянках або на узбіччі. Вона зосереджена на прийнятті спеціальної операційної та цінової політики, що відповідає кожному випадку. Поєднання рішень з планування та управління гарантує, що попит на паркування точно визначається відповідно до фактичних потреб розвитку та умов його роботи, а не відповідно до встановлених загальних стандартів. Це також економить фінансові витрати, час і зручність користувачів, а також зелену інфраструктуру, необхідну для забезпечення екологічного балансу [3].

1.2 Огляд існуючих рішень для виявлення вільних парковок

Розумна система паркування, яка використовує замкнуте телебачення для визначення місця розташування автомобіля на парковці, розроблена в [8 , 9]. Розташування визначається за допомогою розпізнавання пікселів. Розрізнення між наявністю та відсутністю транспортних засобів на парковці визначається встановленням порогового значення для пікселів, а зображення вважаються відтінками сірого . Ця система є більш узгодженою, надійною та точною, і її можна використовувати для отримання інформації про вільні місця на парковці.

Асимільований метод процедур обробки зображень, званий інформаційною системою про зайнятість автомобіля (COINS), пропонується в [10] , щоб отримати інформацію про те, чи припаркований автомобіль, чи вільне місце для паркування. У той же час використання процедур обробки

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

зображень знижує вартість системи. У [11] вільні паркувальні місця ідентифікуються за допомогою магнітного поля Землі. Комп'ютери, які регулюють систему, підключаються до датчиків у цій системі для отримання необхідної інформації.

У [12] пропонується алгоритм для методу паркування, який при реалізації здатний визначати, коли місце для паркування звільнено або зайнято транспортними засобами. Цей алгоритм заснований на радіочастотній ідентифікації. Крім того, водій отримує відомості про вільне паркувальне місце, яке є найближчим до автомобіля в цей момент. Подібно до цього, процедура, яка базується на радіочастотній ідентифікації, запропонована в [13]. Він складається з регулятора ПК воріт та обладнання, пов'язаного з імплантованими воротами. Це гнучка система, так що контролер і обладнання можуть бути змінені в системі.

Алгоритм, який покращує продуктивність існуючої системи розумного паркування, яка базується на хмарних технологіях, запропоновано в [14]. Крім того, IoT використовується для побудови архітектури системи. За допомогою цього алгоритму можна легко визначити вільне місце для паркування, і він автоматизований. Вартість паркування користувача визначається на основі відстані та кількості вільних місць на парковці. Місце для паркування може бути виділено користувачеві за запитом на основі цієї вартості. Користувачеві можна запропонувати іншу стоянку, якщо на запитаній стоянці немає вільного місця. Автори стверджують, що запропонований алгоритм скорочує час очікування користувача та збільшує можливість отримання паркувального місця.

Середовище, яке розглядається для впровадження розумної системи паркування в [15], є міським. Функція витрат водія розраховується на основі того, наскільки ближче користувач до місця призначення та плати за місце для паркування. Ця функція вартості використовується для розподілу та резервування найбільш підходящого паркувального місця для запитуваного користувача. Використовуючи цю процедуру, у кожен момент часу, коли

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

необхідно прийняти рішення, проблема змішаного цілочисельного лінійного програмування (MILP) вирішено. Найкраще підходящий розподіл місця для паркування, який залежить від поточних даних, - це визначення MILP. Ця інформація зазначається під час прийняття рішення в наступній інстанції. Забезпечується уникнення зіткнень у резервуванні ресурсів, а також гарантується, що користувачу з функцією найвищої вартості буде виділено лише місце для паркування. Доведено, що час ідентифікації паркувального місця та плата за паркувальний майданчик зменшуються. У той же час показано, що запропонований алгоритм успішно використовує простір для паркування.

Багато досліджень проводиться в системі розумного паркування, де багатошарова [16], parkagent [17] зажим [18] і parksim [19] є системою, запропонованою в початковий період досліджень у цій галузі. У більшості дослідницьких робіт різні місця паркування повідомляються користувачеві заздалегідь, коли запитане місце паркування заповнено. Електронне паркування – це вдосконалена система, яка повідомляє користувача про деталі паркування заздалегідь, коли робиться запит, а також допомагає зарезервувати місце паркування [20]. Інтернет допомагає користувачеві визначити місце для паркування. Транспортний засіб сканується на в'їзді до місць паркування, щоб визначити, чи зарезеровано транспортним засобом слот чи ні, а також перевіряє, чи сплатив користувач місце для паркування чи ні [21].

Комбінація двох технологій ультразвукового та радарного виявлення використовується для реалізації системи розумного паркування в розвитку розумних міст [22]. Основна мета цього запропонованого підходу полягає в тому, щоб побудувати систему паркування, яка є самокерованою, звести до мінімуму затори, забруднення повітря, час очікування користувачів, дозволяючи їм легко ідентифікувати місце для паркування також у години напруженої роботи. У цьому запропонованому підході до системи паркування буде зменшено участь людини. Модуль Wi-Fi використовується для надсилання всієї інформації в хмару для зберігання. Незаповнене

						ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			19

паркувальне місце ідентифікується користувачем за допомогою інтерфейсу. Заповненість і вільність окремого паркувального місця на парковочній зоні постійно контролюється системою, запропонованою в [22]. Система подає попереджувальний сигнал, коли кількість транспортних засобів на парковці та в дорозі перевищує загальну кількість паркувальних місць на парковці. Цю ж систему можна використовувати для паркування кінотеатрів, торгових центрів, університетів тощо та забезпечити ефективну та економічно вигідну процедуру паркування.

Розумна система паркування на основі бронювання запропонована в [23]. Ця система дозволяє користувачам ефективно ідентифікувати незайняте місце для паркування та резервувати ідентифіковане місце. Паркувальні зони оснащені датчиками, які сповіщають систему про вакантність і заповненість слотів у паркувальній зоні, таким чином змінюючи можливість бронювання. Ця кіберфізична система доступна користувачам за допомогою мобільних телефонів, ноутбуків або будь-яких інших мобільних пристроїв.

Основною метою системи паркування, розробленої в [24], є ефективне використання паркувального простору. Відстань між транспортними засобами зменшується, але без компромісів у тому, що водієві зручно виходити з автомобіля. Два варіанти систем паркування обговорюються в [24]. Вони бувають автоматичними і ручними. Система паркування, яка є ручною, є дорожчою в порівнянні з системою паркування, яка є автоматичною. Інтернет речей є найбільш підходящою технологією для впровадження розумної системи паркування, і в цьому документі використовується рівень 1 IoT.

Різні пропозиції різних дослідників щодо системи паркування обговорюються та аналізуються для визначення відповідних переваг і недоліків у [25]. Також автори розробили розумну систему паркування з використанням IoT. Користувачі можуть постійно спостерігати за порожніми місцями на парковках за допомогою мобільних телефонів або ноутбуків. У той же час запропонована система веде себе також як точка доступу.

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

Розумна система паркування на основі IoT запропонована в [26] . Три різні технології, такі як бездротові сенсорні мережі , процедури комп'ютерного зору та android , використовуються для перевірки вільного місця на паркувальних зонах, ідентифікації прибуття автомобіля на паркувальну зону та інтерфейсу для користувача відповідно. Користувачі можуть визначити вільні місця на парковці, а оплату також можна здійснити за допомогою програми, розробленої за допомогою Android.

Як SDN підходить для VANET досліджено в [35] . Проаналізовано методи залежно від кластерів, геокастинг з використанням придорожного блоку, трансляція, VANET з використанням хмари, SDN і туманного обчислення . Спостерігається, що буде забезпечена хороша якість обслуговування, затримка буде зменшена, а неоднорідність ефективно обробляється у VANET за допомогою SDN. Автори також пояснили проблеми та проблеми, які виникають під час інтеграції SDN і VANET. Ця стаття забезпечує основу, яка допомагає у виконанні досліджень у цій галузі [35] .

Знайти місце для паркування стає важко, оскільки кількість транспортних засобів значно зростає, що збільшує трафік. iERS – це система паркування, що використовує Інтернет речей (IoT), запропонована в [36] , яка допомагає знайти місце для паркування поблизу, яке є вільним. Цей метод також забезпечує навігацію до ідентифікованого слота.

Новий протокол гібридного гетерогенного вилуговування (HNE-LEACH) для бездротових сенсорних мереж у [37] . Автомати навчання також використовуються для підвищення продуктивності запропонованого алгоритму. Порогові значення відстані між вузлами і базовою станцією (BS) визначають, чи буде вузол безпосередньо спілкуватися з BS або опосередковано спілкуватися з BS.

Автоматизоване машинне навчання осмислюється, його розділи та відповідні методології досліджуються в [38] . Він застосовується в контексті галузей і пояснюється. Автори в [39] досліджували різні методології, які

стосувалися процесу забезпечення та процесу планування громадського транспорту, і надали стислий огляд, щоб зробити його придатним для індійських дорожніх обмежень. Процедура планування транспортних засобів на основі інформації в реальному часі, зібраної за допомогою GPS і датчиків, запропонована в [40]. Машинне навчання використовується для покращення процесу планування. Наявність місць, вибір маршруту, вартість проїзду використовуються в рамках таксономії громадського транспорту в [41]. Різні дослідники використовували автомати навчання для підвищення продуктивності в різних сферах [28], [29], [30], [31], [32], [33], [34].

1.3 Вибір перспективного шляху і постановка задачі дослідження

Більшість водіїв стикається з проблемою пошуку вільного паркомісця, особливо в густонаселених районах міст. Це призводить до великих заторів на дорогах та витрачання часу на пошук парковки.

Створення веб-застосунку для дистанційного виявлення вільних паркомісць може значно спростити життя водіям і зменшити транспортні затори. Використання інтелектуальних методів для виявлення вільних паркомісць дозволить створити ефективний та зручний для водіїв інструмент для швидкого та точного пошуку вільного паркомісця.

Також важливим аспектом є збереження часу та зменшення витрат на паливо водіїв, а також зменшення кількості викидів шкідливих речовин в атмосферу, що в свою чергу сприятиме поліпшенню екологічної ситуації в містах.

Отже, розробка веб-застосунку дистанційного інтелектуального виявлення вільних паркомісць є дуже актуальною і має великий потенціал для поліпшення мобільності та екології в містах.

Метою дипломної роботи є розробка веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць у великих містах з метою

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

підвищення ефективності паркування та зменшення трафіку на дорогах. Для досягнення мети потрібно вирішення такі завдань:

1. визначення проблеми пошуку паркомісць у великих містах
2. аналіз існуючих рішень для виявлення вільних парковок
3. провести моделювання функціональної та структурної складових веб-застосунку
4. розробити алгоритм дистанційного виявлення вільних паркомісць
5. розробити підключення до відео потоку
6. провести навчання моделі на основі глибокого навчання з підкріпленням методом градієнтного спуску
7. розробити веб-інтерфейс застосунку для дистанційного інтелектуального виявлення вільного паркомісця.

Завдання 1-2 розглянуто в параграфах 1.1 - 1.4 відповідно, а виконання завдань 3-7 представлені у параграфах 2.1-3.3.

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

2 РОЗРОБКА ФУНКЦІОНАЛУ ТА АЛГОРИТМІВ ВЕБ-ЗАСТОСУНКУ

2.1 Моделювання функціональної та структурної складових веб-застосунку

На рисунку 2.1 зображені усі процеси у короткім текстовім описі, побудовані на основі діаграм Насс-Шнейдермана.



Рис. 2.1 Вигляд діаграми Насс-Шнейдермана

На рисунках 2.1 та 2.2 зображені діаграми переходів станів, які описують відносини між вхідними та вихідними керуючими потоками.



Зм.	Арк.	№ докум.	Підпис	Дата

ДП.КН.8091455.069.ПЗ

Арк.

24

Рис. 2.1. Вигляд діаграми переходів станів для опису відносин користувача та керуючими потоками

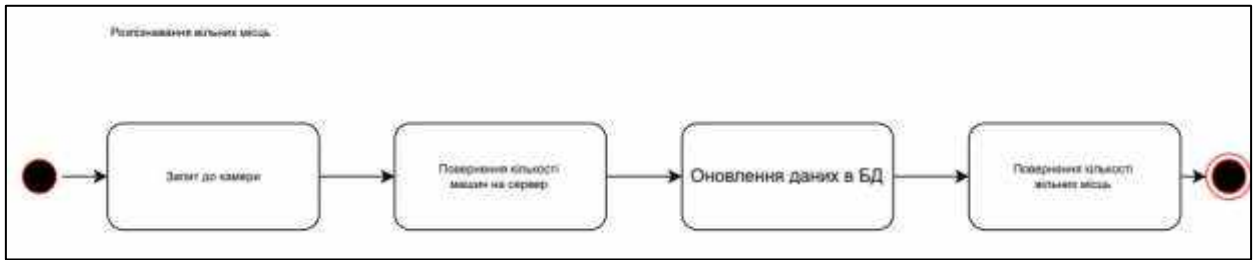


Рис. 2.2. Вигляд діаграми переходів станів для опису процесу розпізнавання вільних місць

На рисунку 2.3 зображена функціональна діаграма взаємозв'язку функцій розроблюваного програмного забезпечення. Діаграма дозволяє виявити основні функції і складові частини проектованої системи і, по можливості, виявити та усунути істотні помилки.

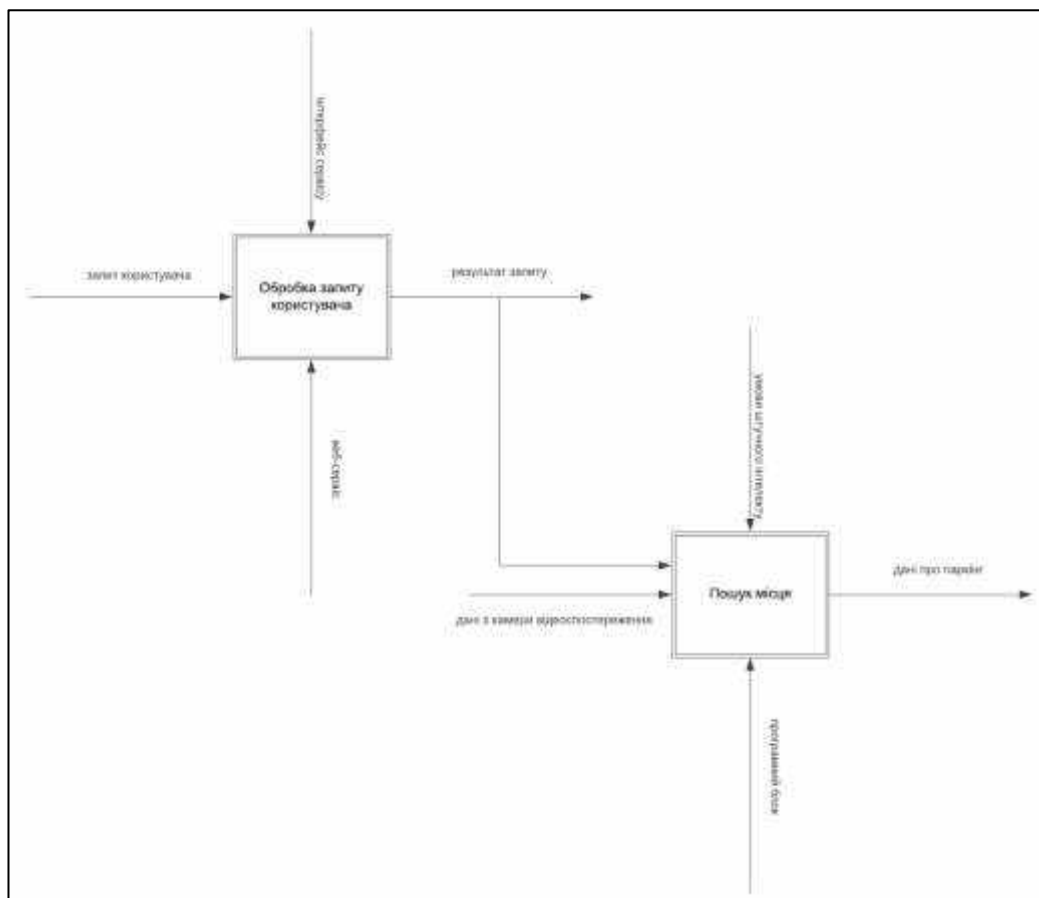


Рис. 2.3. Вигляд функціональної діаграми процесів веб-сервісу

Зм.	Арк.	№ докум.	Підпис	Дата

що всі атрибути кожної таблиці функціонально залежні від їхнього первинного ключа у ній, і таблиці не мають складних потенційних ключів.

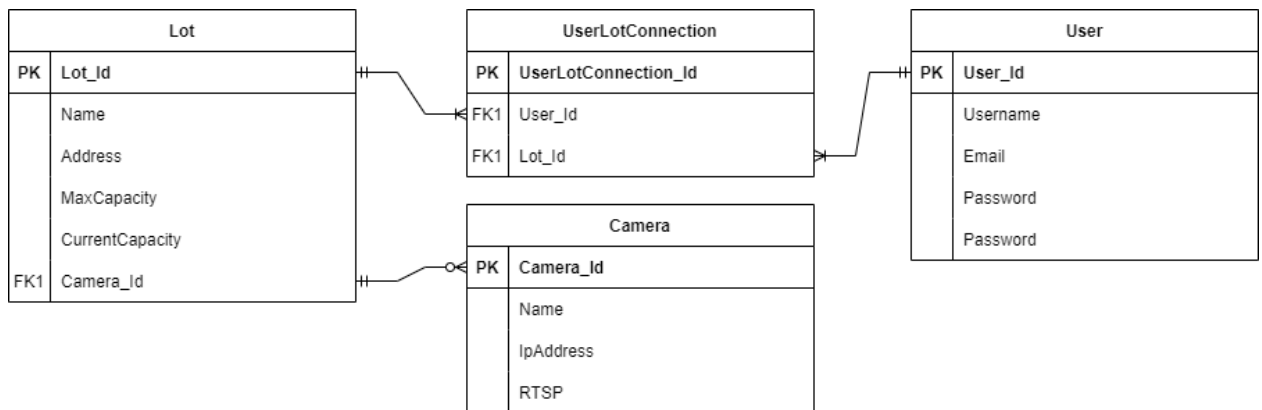


Рис. 2.5. Вигляд діаграми «сутність-зв'язок» для веб-сервісу

2.2 Алгоритм дистанційного виявлення вільних паркомісць

Структурною називають схему, що відображає склад і взаємодію по управлінню частин майбутнього програмного забезпечення.

Структурні схеми пакетів програм не інформативні, оскільки організація програм в пакети не передбачає передачі управління між ними. Тому структурні схеми розробляють для кожної програми пакету окремо, а список програм пакету визначають аналізуючи функції (Рис.2.6).

Функціональна схема – це діаграма, яка зображує процес, систему або алгоритм комп'ютера. Вони широко використовуються в різних сферах для документування, вивчення, планування, покращення та передачі часто складних процесів у чітких і зрозумілих діаграмах (Див.дод.А).



Рис. 2.6. Зображення структурної схеми для знаходження вільного місця на стоянці

Метод покрокової деталізації реалізує низхідний підхід до програмування і припускає покрокову розробку алгоритму.

Нижче продемонстрований метод покрокової деталізації користування сервісом.

Рівень 1: Веб-сервіс

вхід на веб-сервіс

Якщо користувач зареєстрований

реєстрація користувача

автоматична автентифікація користувача

Інакше

автентифікація користувача

вхід на сторінку з стоянками

Цикл-поки стоянка не знайдена

ввід назви стоянки

Кінець-цикл

перегляд інформації про стоянку

Кінець.

Деталізація 1.1. Реєстрацію користувача можна деталізувати на псевдокодї таким чином:

Реєстрація користувача:

Зм.	Арк.	№ докум.	Підпис	Дата

Цикл-поки: користувач не знайдений у БД
ввід даних користувача

Кінець-цикл

додавання даних користувача в БД

Деталізація 1.2. Автентифікацію користувача можна деталізувати на псевдокоді таким чином:

Автентифікація користувача:

Цикл-поки: дані користувача не правильні
ввід даних користувача

Кінець-цикл

авторизація користувача

Для необхідності можна буде додати й інші рівні, які будуть вирішувати питання роботи процесів додавання даних у БД та розпізнавання машин на стоянці.

Методика структурних карт використовується на етапі проектування ПЗ для того, щоб продемонструвати, яким чином програмний продукт виконує системні вимоги, використовуючи структурні карти Константайна (Constantine), призначені для опису відносин між модулями (див.дод.Б).

2.3 Метод класифікації індивідуальних паркувальних місць

Класифікація індивідуальних паркувальних місць є важливим етапом розробки веб-застосунку дистанційного інтелектуального виявлення вільного паркомісця. Цей етап передбачає розподіл паркомісць на категорії залежно від різних факторів, таких як розміри, розташування, тип покриття та доступність для людей з особливими потребами. Класифікація місць дозволяє забезпечити оптимальне використання парковки, забезпечити комфорт користувачів та зменшити час пошуку вільного місця.

Правильна класифікація паркомісць допомагає зменшити витрати часу та грошей на пошук вільного паркомісця, а також зменшити загальний час на пошук паркомісця великими групами людей або на затишних вуличках. Крім

					ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

того, класифікація дозволяє забезпечити безпеку користувачів, зокрема людей з обмеженими можливостями, та зменшити ризик виникнення дорожньо-транспортних пригод через неправильне паркування. Отже, правильна класифікація індивідуальних паркомісць є ключовим етапом в розробці веб-застосунку дистанційного інтелектуального виявлення вільного паркомісця.

Існує кілька методів класифікації індивідуальних паркувальних місць, які використовуються для їх визначення та подальшої інформаційної обробки.

Основними методами класифікації є:

1. Метод на основі відеоаналізу: полягає у використанні відеокамер для визначення стану парковки. Цей метод використовує різні алгоритми обробки відео та машинного навчання для визначення наявності вільного місця на парковці.
2. Метод на основі датчиків: використовується для вимірювання різних параметрів (наприклад, тиску, потужності сигналу, температури тощо), які допомагають визначити наявність вільного місця на парковці.
3. Метод на основі GPS: полягає у використанні системи GPS для визначення місцезнаходження автомобілів та вільних парковочних місць. Цей метод є досить точним, але може бути неефективним у густонаселених місцях з високою забудовою.
4. Метод на основі аналізу звуку: використовується для визначення звукових сигналів, які виділяються автомобілями під час паркування. Цей метод може бути корисним у тих випадках, коли інші методи не є ефективними.

Для поставленої задачі використовуються методи на основі відеоаналізу.

Метод на основі відеоаналізу для класифікації індивідуальних паркувальних місць використовує відеокамери для захоплення зображення вільних та зайнятих парковочних місць. Відеоаналіз використовується для визначення наявності транспортних засобів на парковці, що дозволяє класифікувати парковочні місця як вільні або зайняті.

					ДП.КН.8091455.069.ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Метод використовує алгоритми комп'ютерного зору, які дозволяють автоматично визначати наявність транспортних засобів на парковці та розпізнавати їх тип. Для цього використовуються методи машинного навчання, такі як нейронні мережі, дерева рішень, градієнтний спуск та інші.

Градієнтний спуск (GD) — це ітеративний алгоритм оптимізації першого порядку, який використовується для знаходження локального мінімуму/максимуму заданої функції. Цей метод зазвичай використовується в машинному навчанні (ML) і глибокому навчанні (DL), щоб мінімізувати функцію витрати/втрати (наприклад, у лінійній регресії).

Інтуїтивно це нахил кривої в даній точці в заданому напрямку.

У випадку одновимірної функції це просто перша похідна у вибраній точці. У випадку функції багатьох змінних це вектор похідних у кожному головному напрямку (вдоль осей змінних). Оскільки нас цікавить лише нахил вздовж однієї осі, а інші нас не цікавлять, ці похідні називають частковими похідними.

Градієнт для n -вимірної функції $f(x)$ у заданій точці p визначається таким чином:

$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_n}(p) \end{bmatrix}$$

Перевернутий трикутник є так званим символом набла, і ви читаєте його «del». Щоб краще зрозуміти, як це обчислити, давайте зробимо обчислення вручну для прикладу двовимірної функції нижче.

$$f(x) = 0.5x^2 + y^2$$

									ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата						31

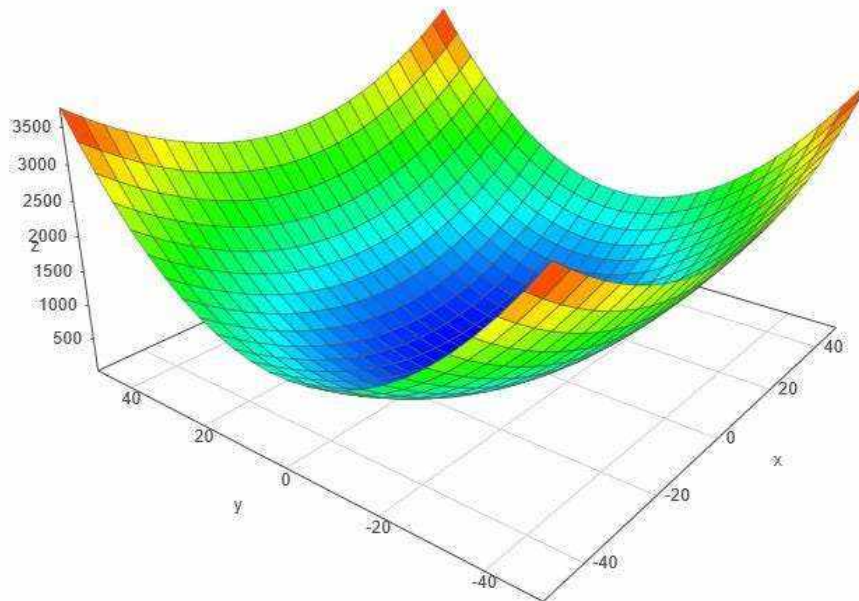


Рис.2.9. 3D сюжет градієнтного спуску

Припустимо, що нас цікавить градієнт у точці $p(10,10)$:

$$\frac{\partial f(x, y)}{\partial x} = x, \quad \frac{\partial f(x, y)}{\partial y} = 2y$$

отже, отже:

$$\nabla f(x, y) = \begin{bmatrix} x \\ 2y \end{bmatrix}$$

$$\nabla f(10, 10) = \begin{bmatrix} 10 \\ 20 \end{bmatrix}$$

Переглядаючи ці значення, ми робимо висновок, що нахил вдвічі крутіший уздовж осі y .

Алгоритм градієнтного спуску ітеративно обчислює наступну точку, використовуючи градієнт у поточній позиції, масштабує її (за швидкістю навчання) і віднімає отримане значення з поточної позиції (робить крок). Він віднімає значення, тому що ми хочемо мінімізувати функцію (щоб максимізувати це було б додаванням). Цей процес можна записати так:

$$p_{n+1} = p_n - \eta \nabla f(p_n)$$

Існує важливий параметр η , який масштабує градієнт і таким чином контролює розмір кроку. У машинному навчанні це називається швидкістю навчання та має сильний вплив на продуктивність.

- Чим менша швидкість навчання, тим довше GD сходиться або може досягти максимальної ітерації до досягнення оптимальної точки
- Якщо швидкість навчання надто висока, алгоритм може не сходитися до оптимальної точки (скакати) або навіть повністю розходитися.

Таким чином, кроки методу градієнтного спуску такі:

1. вибрати початкову точку (ініціалізація)
2. обчисліть градієнт у цій точці
3. зробити масштабований крок у протилежному напрямку до градієнта (мета: мінімізувати)
4. повторюйте пункти 2 і 3, доки не буде виконано один із критеріїв:
 - досягнуто максимальної кількості ітерацій
 - розмір кроку менший за допуск (через масштабування або малий градієнт).

Завдяки використанню відеоаналізу, метод дозволяє забезпечити високу точність класифікації парковочних місць, оскільки він не залежить від фізичного контакту з автомобілем, що забезпечує більш точне визначення стану парковки. Крім того, метод на основі відеоаналізу дозволяє вести статистику зайнятості парковок, що дозволяє відслідковувати тенденції використання парковочних місць та планувати їх ефективне використання.

					ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Підключення до відео потоку

Підключення до відкритого відеопотоку можна здійснити за допомогою API (Application Programming Interface). Це інтерфейс, який надається розробниками для взаємодії з їхніми додатками та сервісами.

Для підключення до відеопотоку через API необхідно спочатку зареєструватися на сайті, який надає цей сервіс та отримати API-ключ. Далі потрібно скласти запит до API, який містить URL відеопотоку та параметри підключення.

Наприклад, якщо відеопотік знаходиться на сервері за адресою "https://camguide.net/usa/", то запит до API може виглядати наступним чином:

```
bashCopy code
```

```
https://camguide.net/usa//api/v1/video?key=API_KEY&format=mp4&resolution=720p
```

де **API_KEY** - це ключ, який вам надали при реєстрації, **format** - формат відео (наприклад, mp4, avi, flv тощо), **resolution** - розширення відео (наприклад, 720p, 1080p тощо).

Отриманий запит необхідно передати до API-сервера за допомогою HTTP-запиту з методом GET або POST, залежно від того, яким методом передбачається робота з API.

Далі проводимо підключення та виведення камер у розробленому веб порталі.

На першому етапі підключаємо необхідні бібліотеки. Даний код відноситься до веб-додатку на платформі ASP.NET Core і включає в собі декілька залежностей, таких як **GetParked.Models**, **Microsoft.AspNetCore.Authorization**, **Microsoft.AspNetCore.Http**,

									ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата						34

перевіряє, чи переданий параметр `id` є дійсним існуючим ідентифікатором камери в базі даних. Якщо `id` є недійсним, метод повертає об'єкт `NotFoundResult`, що вказує на помилку 404. Інакше, метод витягує з бази даних запис камери з відповідним ідентифікатором разом з додатковою інформацією про парковку, пов'язану з цією камерою. Завершується метод поверненням відображення з відповідним записом камери та даними про парковку в шаблоні `Details`.

```
// GET: Cameras/Details/5
public async Task<IActionResult> Details(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var lot = await _context.Cameras
        .Include(l => l.Lot)
        .FirstOrDefaultAsync(m => m.Id == id);
    if (lot == null)
    {
        return NotFound();
    }

    return View(lot);
}
```

Далі код відповідає за створення нової камери і повертає форму для її заповнення. У методі `Create()` контролера `Cameras`, спочатку встановлюється значення `ViewData["Lot"]` як `SelectList`, який містить `Id` та назву парковки, для якої камера не призначена. Потім метод повертається до виду `Create.cshtml`.

Коли користувач заповнює форму та надсилає її, метод `POST Create` обробляє дані форми та перевіряє їх на валідність. Якщо дані є валідними, то камера додається до бази даних та користувач перенаправляється на головну сторінку. В іншому випадку, користувач залишається на сторінці створення камери разом з повідомленням про помилку.

```
// GET: Cameras/Create
public IActionResult Create()
{
    ViewData["Lot"] = new SelectList(_context.Lots.Where(l => l.Camera == null), "Id", "Name");
    return View();
}

// POST: Cameras/Create
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create(Camera camera)
{
    if (ModelState.IsValid)
    {
        _context.Add(camera);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(camera);
}
```

Визначення контролера для редагування об'єкту "Camera". У методі "Edit" перевіряється, чи існує камера з вказаним ідентифікатором, якщо так, то відображається форма для редагування камери, де відображається список парковок для вибору. Після внесення змін до форми та натискання на кнопку "Зберегти", зміни зберігаються в базі даних та відбувається перенаправлення на сторінку "Index". У разі помилки введення даних, знову відображається форма для редагування з повідомленням про помилки.

```
// GET: Cameras/Edit/5
public async Task<IActionResult> Edit(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var lot = await _context.Cameras
        .Include(c => c.Lot)
        .FirstOrDefaultAsync(l => l.Id == id);
    if (lot == null)
    {
        return NotFound();
    }
    ViewData["Lot"] = new SelectList(_context.Lots, "Id", "Name");
    return View(lot);
}

// POST: Cameras/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(int id, Camera camera)
{
    if (id != camera.Id)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(camera);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!CameraExists(camera.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    return View(camera);
}
```

Далі пропишемо частину контролера управління камерами веб-додатку.

Перша частина коду - GET запит на видалення камери з відповідним id. Якщо id не задано або не знайдено камеру з відповідним id, то повертається сторінка з помилкою NotFound. Якщо ж камера знайдена, то повертається сторінка з інформацією про цю камеру.

Друга частина коду - POST запит на видалення камери з відповідним id. Якщо id збігається з id камери, то проводиться спроба видалення цієї камери з бази даних. Якщо видалення успішне, то користувач переадресовується на головну сторінку (Index). Якщо ж виникає помилка, то виводиться сторінка з помилкою.

У кінці коду є приватна функція CameraExists, яка перевіряє, чи існує камера з відповідним id у базі даних. Якщо така камера знайдена, повертається true, якщо ні - false. Ця функція використовується для перевірки наявності камери з відповідним id перед видаленням камери.

```
// GET: Cameras/Delete/5
public async Task<IActionResult> Delete(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var camera = await _context.Cameras
        .FirstOrDefaultAsync(m => m.Id == id);
    if (camera == null)
    {
        return NotFound();
    }

    return View(camera);
}

// POST: Cameras/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    var camera = await _context.Cameras.FindAsync(id);
    _context.Cameras.Remove(camera);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

private bool CameraExists(int id)
{
    return _context.Cameras.Any(e => e.Id == id);
}
```

На завершення представимо клас **Camera**, який відповідає за моделювання відомостей про камеру. У класі визначено наступні поля:

									ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата						38

- **Id**: це унікальний ідентифікатор кожної камери, який генерується автоматично при додаванні нової камери в базу даних.
- **IpAddress**: це IP-адреса камери.
- **RTSP**: це URL-адреса, яка використовується для отримання відеопотоку з камери.
- **LotId**: це ідентифікатор парковки, до якої прив'язана камера.
- **Lot**: це об'єкт парковки, до якої прив'язана камера.

У класі також використовуються атрибути з простору імен **System.ComponentModel.DataAnnotations** для визначення правил валідації даних, таких як обов'язковість введення певних полів та обмеження їх довжини.

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace GetParked.Models
{
    public class Camera
    {
        public int Id { get; set; }
        public string IpAddress { get; set; }
        public string RTSP { get; set; }

        public int? LotId { get; set; }
        public Lot Lot { get; set; }
    }
}
```

3.2 Навчання моделі на основі глибокого навчання з підкріпленням методом градієнтного спуску

Найкращим варіантом для навчання моделі на основі глибокого навчання з підкріпленням методом градієнтного спуску є бібліотека OpenCvSharp.

OpenCvSharp - це .NET-обгортка для бібліотеки комп'ютерного зору OpenCV (Open Source Computer Vision Library). Вона дозволяє розробникам .NET створювати програми з використанням функцій комп'ютерного зору, які

									ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата						39

надає OpenCV, без необхідності вивчати C++ або мати досвід у використанні оригінальної бібліотеки OpenCV.

OpenCVSharp реалізує інтерфейс до OpenCV за допомогою механізму платформи-незалежного .NET-виклику, який дозволяє розробникам взаємодіяти з OpenCV-функціями за допомогою об'єктів .NET, таких як Mat (матриця), Image і Window.

Ця бібліотека дозволяє розробникам .NET створювати різноманітні застосунки, які використовують функції комп'ютерного зору, такі як виявлення облич, виявлення руху, визначення об'єктів і багато іншого. OpenCVSharp надає простий і зручний інтерфейс для взаємодії з OpenCV і дозволяє розробникам .NET створювати програми з використанням функцій комп'ютерного зору, які ефективно працюють з великою кількістю даних.

OpenCVSharp.dnn - це модуль в бібліотеці OpenCVSharp, який надає можливість використовувати глибокі нейронні мережі (deep neural networks) для обробки зображень та відео. Цей модуль дозволяє завантажувати популярні архітектури нейронних мереж, такі як YOLO, SSD, Faster R-CNN, та інші, та використовувати їх для класифікації, детектування та відстеження об'єктів на зображеннях та відео.

OpenCVSharp.dnn використовує методи машинного навчання, зокрема глибоке навчання (deep learning), для навчання моделей розпізнавання об'єктів на зображеннях та відео. Для навчання моделей використовуються великі набори даних, які містять зображення з позначеними об'єктами, та використовується алгоритм зворотного поширення помилок (backpropagation), щоб оптимізувати параметри моделі для забезпечення точної класифікації та детекції об'єктів на нових зображеннях.

YOLO (You Only Look Once) - це архітектура глибокого навчання для об'єктного виявлення в реальному часі. Вона використовує нейронні мережі для виявлення об'єктів в зображенні та призначення їм класів. Особливістю YOLO є те, що вона працює з відеопотоками в реальному часі, що дозволяє їй розпізнавати об'єкти в режимі онлайн без значного збільшення часу обробки.

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

нові об'єкти. Набір даних, який використовувався для виявлення 80 класів об'єктів, відомий під назвою "Coco".

Спочатку зображення, яке вводиться до мережі, поділяється на секції. Візьмемо для прикладу матрицю-сітку 3×3 (Рис.3.2).



Рисунок 3.2 - Приклад матриці-сітки 3×3

Отже, цього зображення було дано формат матриці розміром 3×3. На цьому фото в цілому 9 секцій. Кожна їх має певні параметри. Якщо ми припустимо, що загальна кількість класів, які ми шукаємо на малюнку, дорівнює 3 (імовірно, це будуть людина, автомобіль і літак), то кожна секція матиме загалом 8 параметрів. Але чому саме 8? Тому що кожна секція містить 5 параметрів та три параметри класу.

Щоб коментувати ці параметри, нам потрібно знати, що таке рамки, що обмежують. Під час підготовки навчальних даних ми маємо виділити об'єкт, який хочемо виявити на зображенні. Робимо ми це за допомогою рамок, що обмежують. Як правило, вони є квадратами або прямокутниками, які виділяють певну частину зображення (Рис. 3.3).

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.3 - Приклад розпізнавання

В даному випадку нам потрібно виявити автомобілі, тому ми поміщаємо рамки, що обмежують, навколо всіх автомобілів, присутніх на ньому. Тепер нам необхідно дізнатися значення 5 параметрів у кожній секції у матриці 3×3 . Нижче представлено фотографію окремої секції, в якій є автомобіль (Рис.3.4).



Рисунок 3.4 - Фотографія окремої секції, в якій є автомобіль

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

Червона точка в середині позначає центр рамки, що обмежує. Горизонтальна синя стрілка - це параметр "tx" (відстань між червоною точкою і лівою частиною цієї секції). Вертикальна синя стрілка – це "ty" (відстань між червоною точкою та найвищою частиною секції). Горизонтальна біла стрілка – це ширина обмежувальної рамки по відношенню до секції (параметр tw). Вертикальна біла стрілка позначає висоту рамки по відношенню до секції і вказується як "th".

Параметр "po", також відомий як індекс об'єктності, виражає ймовірність успішного виявлення об'єкта в рамці, що обмежує. І так, ви вгадали, значення 0,99 на першій картинці цієї статті-індекс об'єктності, що вказує на наявність особи в рамці, що обмежує, яка його оточує. Індокси p1, p2 і p3 говорять нам про ймовірність того, що цей об'єкт виявиться людиною, автомобілем або літаком відповідно. Усі 9 секцій, присутніх у матриці 3×3, мають ці 8 параметрів, і вони допомагають алгоритму YOLO точно виявити об'єкт.

YOLO використаємо для обробки відеопотоку з камери та аналізу зображення, щоб визначити, чи є вільне парковочне місце на даний момент. Використання YOLO може забезпечити швидку та ефективну обробку відео, що дозволить визначати вільні парковочні місця в режимі реального часу.

Отже, розглянемо опис реалізації YOLO у визначати вільні парковочні місця в режимі реального часу. І відповідно на першому етапі підключаємо потрібні бібліотеки.

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Net;
using System.Reflection;
using System.Threading;
using OpenCvSharp;
using OpenCvSharp.Dnn;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Firefox;
```

Підключаємо **YoloDetect**, який містить в собі деякі константи та статичні поля.

- **Cfg, Weight, Names** - рядки, які вказують на шляхи до конфігураційного файлу, вагових файлів та файлу з назвами класів для об'єктів, які відтворює нейронна мережа YOLOv3.
- **Location** - рядок, який містить в собі шлях до папки з файлами конфігурації та даними для нейронної мережі.
- **Colors** - статичний масив об'єктів типу **Scalar**, який містить кольори для малювання рамок навколо об'єктів на вихідному зображенні. Для кожного класу випадковим чином обирається один з кольорів.
- **Labels** - статичний масив рядків, який містить назви класів для об'єктів, які відтворює YOLOv3.
- **detectedObjects** - статичний список рядків, який містить назви об'єктів, виявлених на вихідному зображенні.

В коді використовуються директиви **using**, які підключають необхідні простори імен для використання класів та методів з цих просторів.

									Арк.
									45
Зм.	Арк.	№ докум.	Підпис	Дата					

ДП.КН.8091455.069.ПЗ

та порог для Non-maximum Suppression (NMS) відповідно. Наступний блок коду отримує зображення та створює blob зображення, що потрібно для передачі в нейромережу. Після цього код завантажує файл конфігурації та ваговий файл моделі, встановлює оптимальну архітектуру та виконує прямий прохід мережі з вхідними даними. Після виконання прямого проходу надходять результати з усіх вихідних шарів мережі, які оброблюються функцією **GetResult()**. Останній блок коду показує вихідне зображення з відміченими рамками, що вказують на виявлені об'єкти, та зберігає його на диск.

```
#region forward model
Stopwatch sw = new Stopwatch();
sw.Start();

net.Forward(outs, outNames);

sw.Stop();
Console.WriteLine($"Runtime:{sw.ElapsedMilliseconds} ms");
#endregion

//get result from all output
GetResult(outs, org, threshold, nmsThreshold);

org.Resize(org.Size(), 800, 600);

// Show window with objects' boxes (ONLY FOR DEBUGGING)
using (new Window("frame.tw", org))
{
    Cv2.WaitKey();
}

org.SaveImage("../Content/page_screenshot_with_boxes.jpg");
}
```

Створюємо функцією для отримання результатів детекції об'єктів на зображенні. Вхідні параметри включають список матриць виводу моделі детектування, зображення, порог впевненості та порог для non-maximum suppression (NMS), який допомагає уникнути перекриття об'єктів. Функція проходить через кожну матрицю виводу і використовує вивід для отримання координат центру та розміру області обмеження кожного об'єкту. Вона також отримує імовірність, що об'єкт належить до певного класу, та впевненість у цьому результаті. Функція повертає список класів, вірогідності, розмірів та координат областей обмеження, які згенеровано моделлю детектування.

									ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата						48

кількість об'єктів, які були вилучені за допомогою non-maximum suppression, та виводить на зображенні залишені об'єкти.

```
        if (!nms)
        {
            // draw result (if don't use NMSBoxes)
            GetDrawInfo(image, classes, confidence, probability, centerX, centerY, width, height);
            continue;
        }

        //put data to list for NMSBoxes
        classIds.Add(classes);
        confidences.Add(confidence);
        probabilities.Add(probability);
        boxes.Add(new Rect2d(centerX, centerY, width, height));
    }
}

if (!nms) return;

//using non-maximum suppression to reduce overlapping low confidence box
CvDnn.NMSBoxes(boxes, confidences, threshold, nmsThreshold, out int[] indices);

Console.WriteLine($"NMSBoxes drop {confidences.Count - indices.Length} overlapping result.");

foreach (var i in indices)
{
    var box = boxes[i];
    GetDrawInfo(image, classIds[i], confidences[i], probabilities[i], box.X, box.Y, box.Width, box.Height);
}
}
```

Далі отримуємо зображення, клас, ймовірність та координати області об'єкту, що виявлено на зображенні. Якщо клас зображення є "car", "truck" або "bus", метод додає мітку цього об'єкту до списку виявлених об'єктів та намагається відобразити прямокутник, що обведе об'єкт та назву класу. Колір та текст, що відображаються виконуються на основі кольору об'єкту. Якщо кольори об'єкту є темними, текст стає білим, інакше текст стає чорним. Цей метод використовується лише для налагодження, тому що він має високі вимоги до продуктивності та працює дуже повільно.

```
private static void GetDrawInfo(
    Mat image, int classes,
    float confidence, float probability,
    double centerX, double centerY, double width, double height)
{
    var label = $"{Labels[classes]}";
    if (label == "car" || label == "truck" || label == "bus")
    {
        DetectedObjects.Add(label);

        //Setup boxes on objects(ONLY FOR DEBUGGING)
        var xl = (centerX - width / 2) < 0 ? 0 : centerX - width / 2;
        image.Rectangle(new Point(xl, centerY - height / 2), new Point(centerX + width / 2, centerY + height / 2), Colors[classes], 2);
        var textSize = Cv2.GetTextSize(label, HersheyFonts.HersheyTriplex, 0.8, 1, out var baseline);
        Cv2.Rectangle(image, new Rect(new Point(xl, centerY - height / 2 - textSize.Height - baseline),
            new Size(textSize.Width, textSize.Height + baseline)), Colors[classes], CV2.FILLED);
        var textColor = Cv2.Mean(Colors[classes]).Val0 < 70 ? Scalar.White : Scalar.Black;
        Cv2.PutText(image, label, new Point(xl, centerY - height / 2 - baseline), HersheyFonts.HersheyTriplex, 0.8, textColor);
    }
}
```

Зм.	Арк.	№ докум.	Підпис	Дата

ДП.КН.8091455.069.ПЗ

Арк.

50

Цей фрагмент коду містить функцію "GetDetectedData()", яка повертає список розпізнаних об'єктів, зібраних під час виконання функції "GetResult()". Список розпізнаних об'єктів є приватним полем класу і заповнюється під час виконання методу "GetResult()", який розпізнає об'єкти на зображенні. Після виклику "GetResult()", список розпізнаних об'єктів може бути отриманий за допомогою виклику "GetDetectedData()". Після формується база з розпізнаними парковками (рис.3.5).

```
public List<string> GetDetectedData ()  
{  
    ...  
    return detectedObjects;  
}
```



Рисунок 3.5 - Приклади розпізнаних парковок

І в кінці використовуємо функцію, що полягає в очищенні списку розпізнаних об'єктів. Метод отримує доступ до змінної detectedObjects, яка є списком рядків інформації про розпізнані об'єкти, і замінює його на новий порожній список. Це дає можливість розпочати новий процес розпізнавання об'єктів із чистого листа, не додавши до списку нових розпізнаних об'єктів.

```
public void ResetDetectedData ()
{
    detectedObjects = new List<string> ();
}
```

3.3 Веб-інтерфейс застосунку для дистанційного інтелектуального виявлення вільного паркомісця

Веб-застосунок дистанційного інтелектуального виявлення вільного паркомісця є основним компонентом розробленої системи. Він забезпечує збір інформації з камер, виконує обробку даних, аналізує їх і визначає наявність вільних паркомісць.

Застосунок має веб-інтерфейс, який дозволяє користувачам переглядати поточну ситуацію на парковках у режимі реального часу. Він також забезпечує можливість відстежування історії використання паркомісць, створення звітів і аналізу даних.

Для розробки веб-застосунку використовується стек технологій, що включає у себе HTML, CSS, JavaScript, а також фреймворки React та Node.js. Використання цих технологій дозволяє створювати веб-інтерфейс, який працює швидко та має високу ступінь взаємодії з користувачем.

Застосунок повинен бути зручним та інтуїтивно зрозумілим для користувачів, а також містити всі необхідні функції для ефективного виявлення вільних паркомісць. Він повинен бути надійним та стабільним, щоб забезпечити безперебійну роботу системи.

Набір функцій та контент, які задіяні в проекті:

Головна сторінка

Header

- Додому
- Стоянки
- Блог
- Про нас

									ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата						52

- Підтримка
- Вхід\реєстрація

Блок 1

- Назва продукту
- Фотоілюстрація, яка асоціюється з продуктом
- Текст призив до дії
- Клавiша «знайти стоянку»

Сторінка стоянки

- Поле пошуку
- Клавiша «почати пошук»

Таблиця з даними стоянок

- Назва
- Адрес
- Доступ(детальніше)

Сторінка деталі стоянки

- Назва стоянки
- Адрес
- Зайнято місць на даний момент (11)
- Максимальна кількість (25)
- Кнопка: назад до списку

Сторінка про нас

- Фото команди
- Заголовок
- Опис
- Контакти кожної персони

Підтримка

- Текст
- Дані картки

У даному проекті була використана сітка з чотирма колонами. Для уточнення дизайну та розміщення блоків на сайті було створено чотири різні варіанти прототипів (зображення показано на Рис. 3.6). Кожен варіант прототипу мав свої особливості та відмінності, що дозволило здійснити більш точне визначення, які блоки та елементи повинні бути включені на сайті. Після

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

аналізу та порівняння чотирьох варіантів прототипу було вирішено вибрати найбільш підходящий варіант для подальшого розроблення веб-застосунку.



Рисунок 3.6 - Вигляд 4 варіантів розміщення блоків на сайті

Після того, як всі блоки були узгоджені та розміщені на веб-сайті, наступним кроком була інтеграція тексту (Рис. 3.7). З метою забезпечення єдності стилю та легкості сприйняття, був створений текст-концепт, який відображав головні ідеї та концепцію проекту. Цей текст-концепт був інтегрований на веб-сайті та забезпечив чітке відображення всіх основних аспектів та функціональності проекту. Крім того, він допоміг відвідувачам зрозуміти, як саме працює даний веб-застосунок та як він може бути корисним для вирішення конкретних проблем.

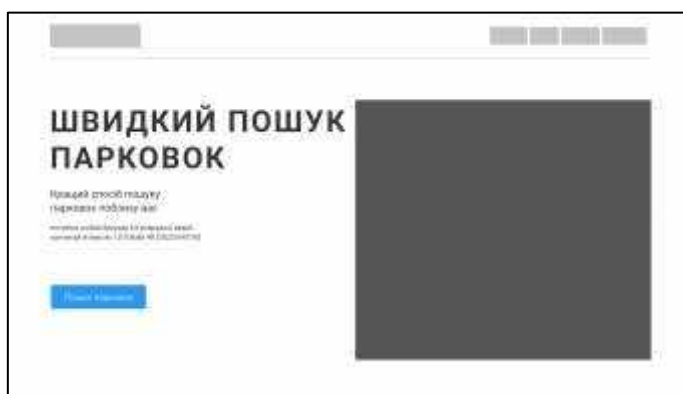


Рисунок 3.7 - Вигляд інтегрованого текст-концепту на початковій веб-сторінці

На етапі деталізації дизайну веб-сторінки додавані додаткові елементи, які забезпечують зручну навігацію для користувачів (Рис. 3.8). Ці елементи можуть включати клавiші, посилання, логотип, іконки та візуальний стиль, такий як колір та типографіка. Наприклад, клавiші можуть допомогти користувачам здійснити певну дію, таку як зареєструватися або увійти в систему. Посилання можуть бути додані, щоб допомогти користувачам переходити на інші сторінки сайту. Логотип і іконки можуть бути використані для ідентифікації бренду та створення знайомого візуального стилю. Крім того, використання певного кольору може змінити загальний вигляд та настрої веб-сторінки, що може вплинути на сприйняття користувачів. Всі ці елементи додаються на етапі деталізації з метою поліпшення дизайну та забезпечення зручної навігації для користувачів.

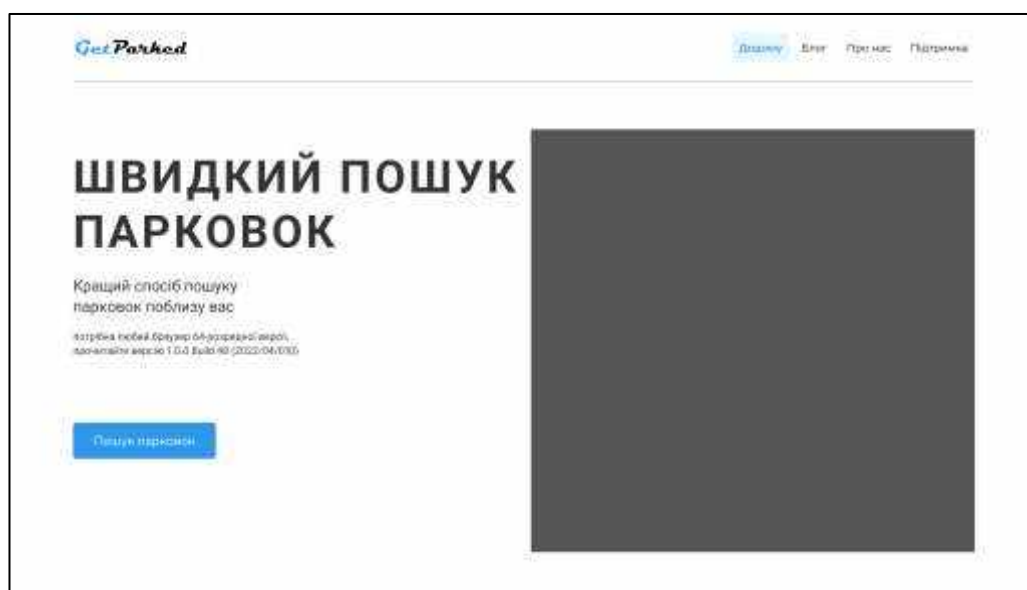


Рисунок 3.8 - Вигляд деталізованої початкової сторінки

На етапі деталізації було виконано остаточний вибір візуального оформлення та створено відповідну макетну модель веб-застосунку (Рис. 3.9). Цей етап є важливим для створення привабливого та зручного інтерфейсу, що покращує користувацький досвід та забезпечує ефективну роботу програмного забезпечення.

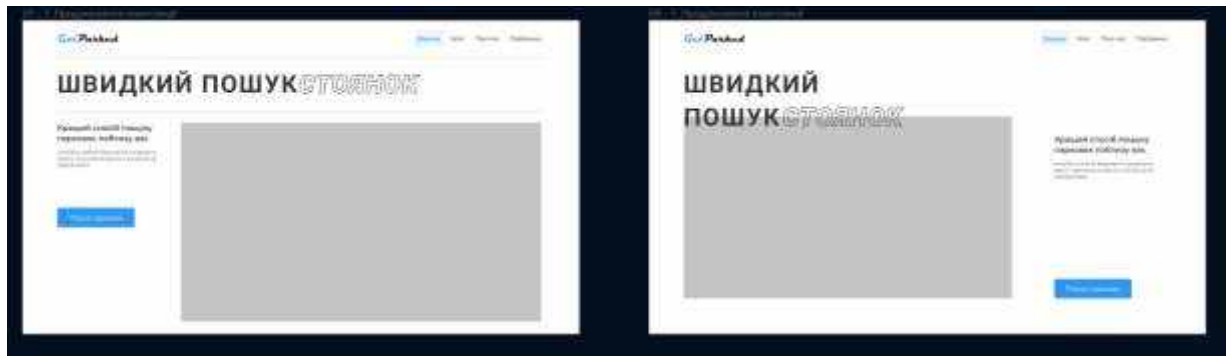


Рисунок 3.9 - Вигляд двох узгоджених варіантів з різним візуальним рішенням

У процесі розробки веб-застосунку було вирішено використати фотоілюстрацію для демонстрації роботи системи і привернення уваги користувачів. Були вибрані кілька варіантів зображень, які мали відповідати тематиці проекту та бути зрозумілими для потенційних користувачів. Після узгодження з замовником, було вибрано оптимальний варіант фотоілюстрації, який був доданий на головну сторінку веб-застосунку (Рис. 3.10). Даний етап є важливим для створення привабливого та зрозумілого продукту, який приверне увагу та забезпечить комфортну роботу користувачів.

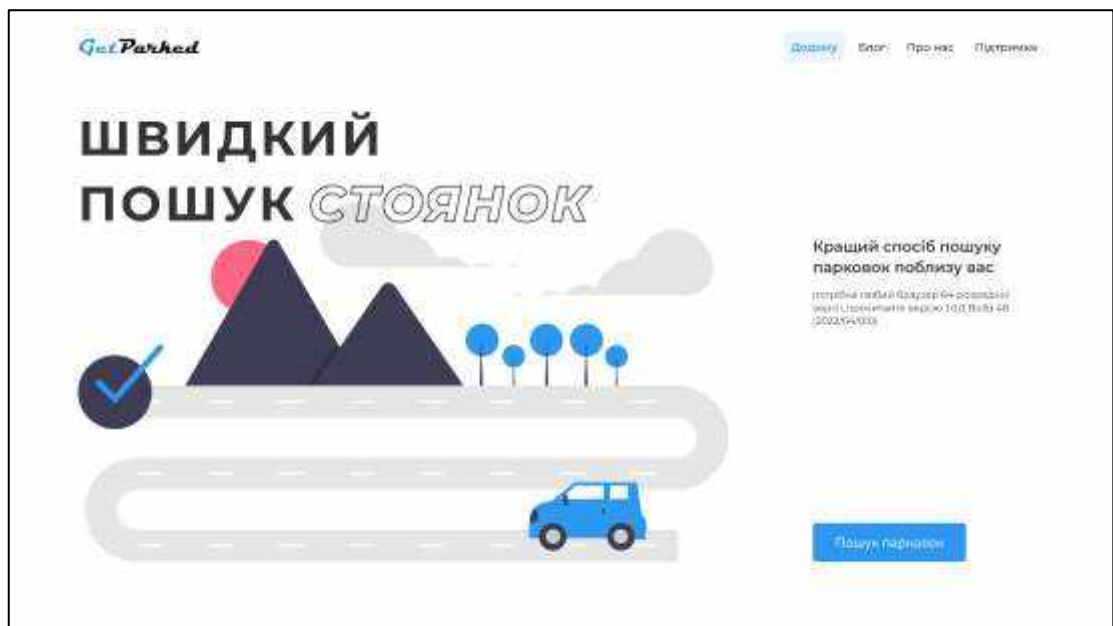


Рисунок 3.10 - Вигляд узгодженого варіанту фотоілюстрації в проекті

Для того, щоб сайт був зручно переглядати на різних пристроях, були використані адаптивні технології. Зокрема, були вибрані наступні розширення: 1600 x 900, 1366 x 900, 1280 x 900, 1000 x 900, 768 x 900, 480 x 900, 360 x 900 (Рис. 3.11, 3.12). Це дозволяє користувачам з різними розмірами екранів насолоджуватися максимально зручним та оптимізованим дизайном сайту.

Наприклад, для користувачів з великими моніторами (1600 x 900 та 1366 x 900) дизайн буде максимально заповнювати екран, та не буде викликати почуття "порожнечі". Для тих, у кого екран менший (наприклад, 768 x 900 або 480 x 900), дизайн сайту буде адаптовуватися до розміру екрану, зберігаючи оптимальну читабельність та зручність користування.

Такі адаптивні розширення дозволяють покращити досвід користувачів та забезпечують максимальний комфорт перегляду сайту на будь-яких пристроях, що є надзвичайно важливим в наш час, коли користувачі активно використовують різні пристрої для перегляду контенту в Інтернеті.

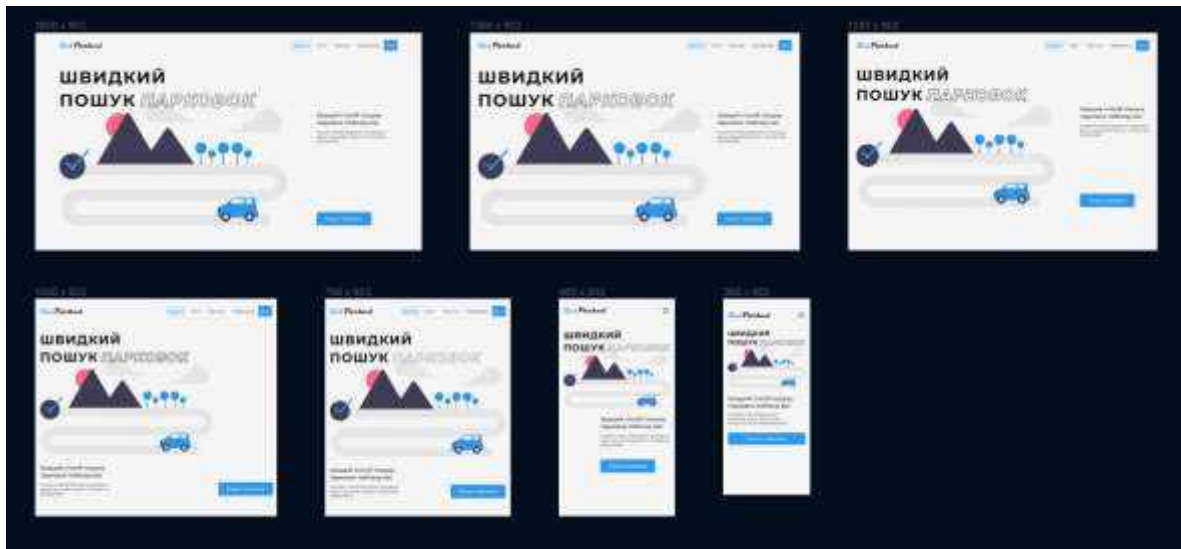


Рисунок 3.11 - Вигляд адаптивів для початкової сторінки веб-сервісу

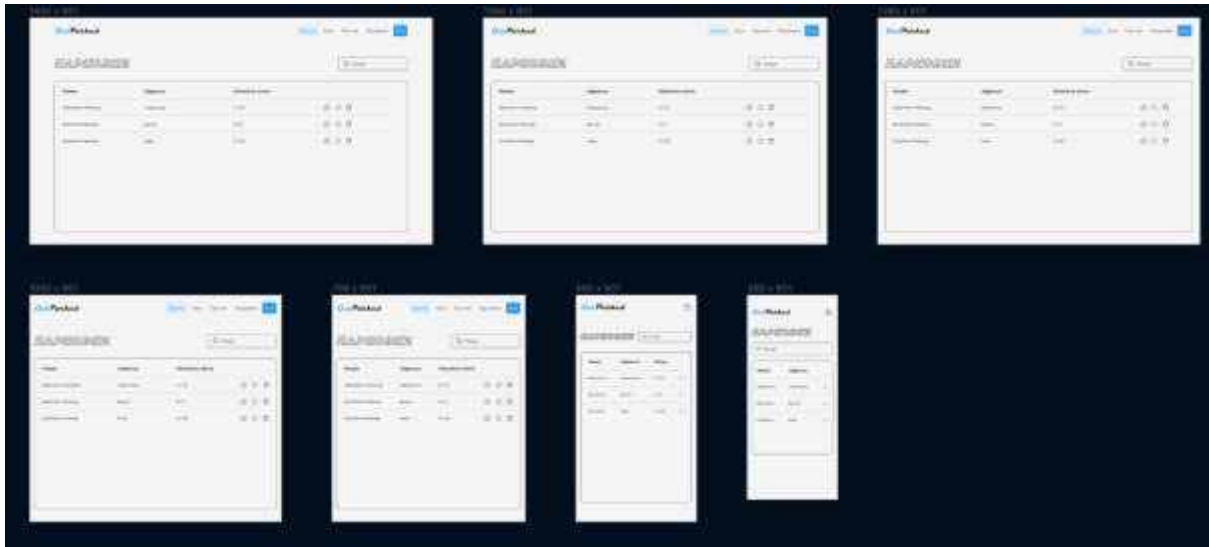


Рисунок 3.12 - Вигляд адаптивів для сторінки пошуку стоянок

UI дизайн сторінки для входу в особистий кабінет має бути зрозумілим та інтуїтивно зрозумілим для користувачів. Основним елементом сторінки є форма входу, яка повинна бути розміщена у центрі сторінки. Форма включає поля для введення логіну та паролю, а також кнопку "Увійти" (Рис. 3.13.).



Рисунок 3.13 - Вигляд UI дизайну сторінки для входу в особистий кабінет

Сторінка реєстрації користувача має такий вигляд: у верхній частині розміщені назва проекту та логотип; у середині сторінки - форма реєстрації з полями для введення інформації: електронна пошта, пароль, підтвердження

паролю. Над формою реєстрації розміщено заголовок "Реєстрація", а під нею - кнопка "Зареєструватися" (Рис. 3.13). У нижній частині сторінки знаходиться посилання на сторінку входу, а також інформація про авторів.

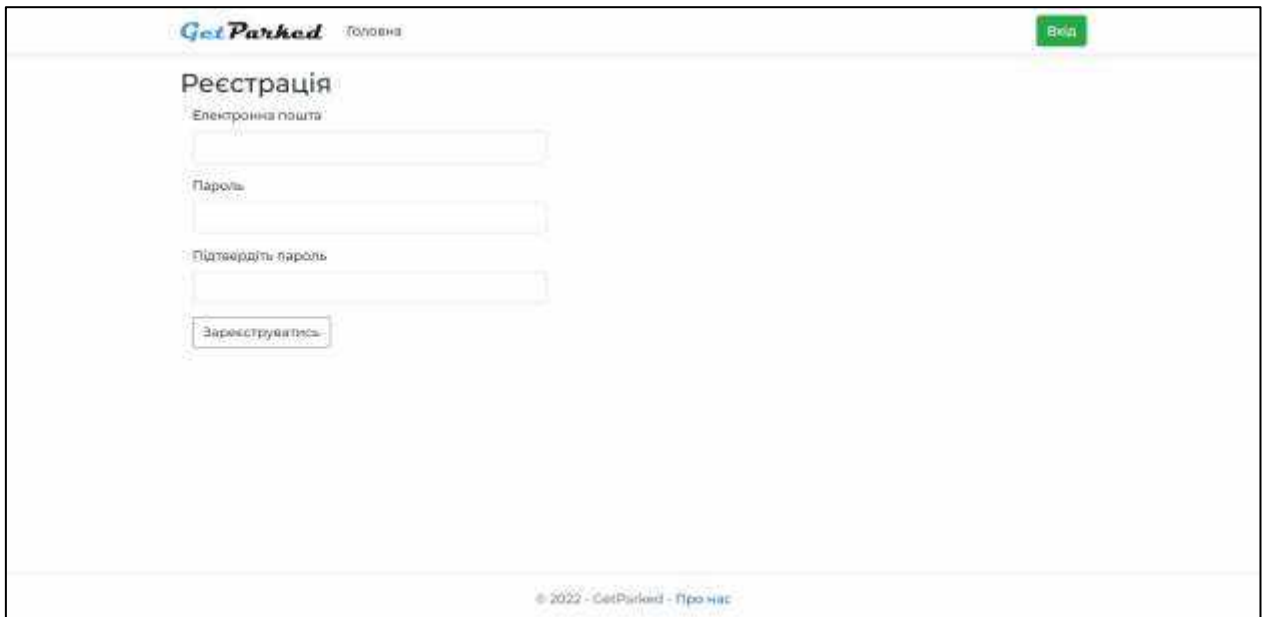


Рисунок 3.13 - Вигляд UI дизайну сторінки реєстрації користувача

Після автентифікації (Рис. 3.14.) користувача головна сторінка має наступний вигляд UI дизайну: у верхній частині розташована навігаційна панель з логотипом та кнопкою "Вихід".

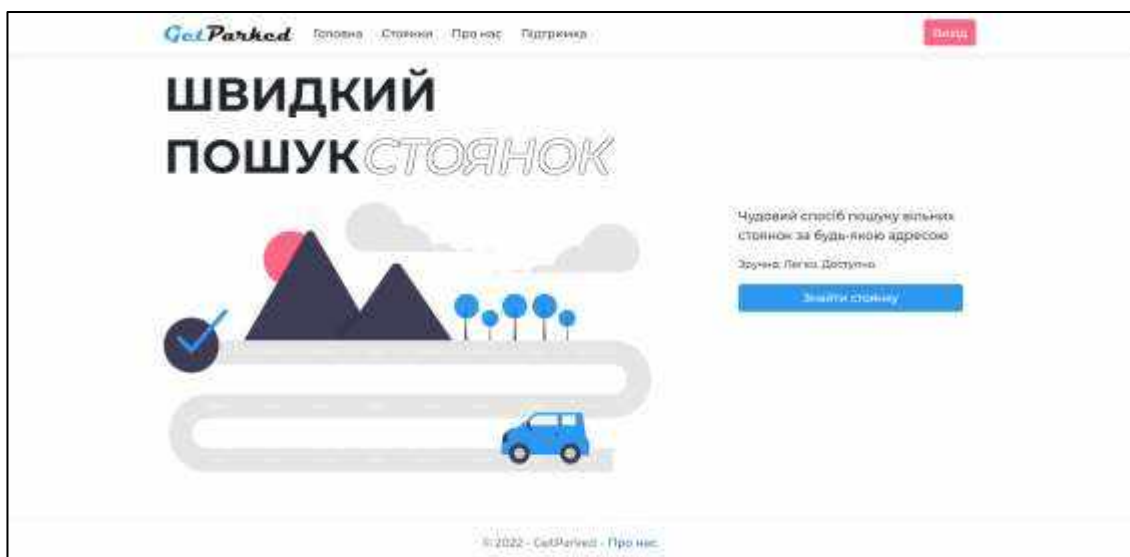


Рисунок 3.14 - Вигляд UI дизайну головної сторінки після автентифікації користувача

На сторінці (Рис. 3.15) пошуку стоянок розміщено таблицю з переліком доступних стоянок, яка містить інформацію про назву стоянки, її адресу та посилання на доступ до камер. Таблиця має простий та зрозумілий дизайн, з лаконічними та чіткими заголовками кожної колонки, що дозволяє швидко та зручно знайти необхідну інформацію.

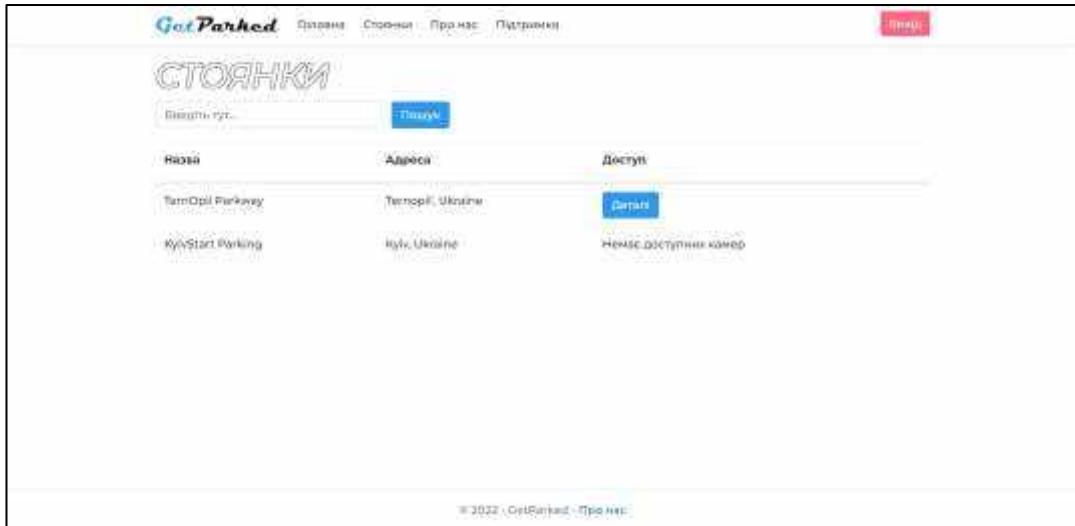


Рисунок 3.15 - Вигляд UI дизайну сторінки пошуку стоянок

Сторінка (Рис. 3.16) перегляду деталей стоянки складається з заголовка, який містить назву стоянки, та блоків з інформацією про адресу та кількість парковок. На сторінці також розміщується таблиця, яка містить детальну інформацію про стан кожної парковки.

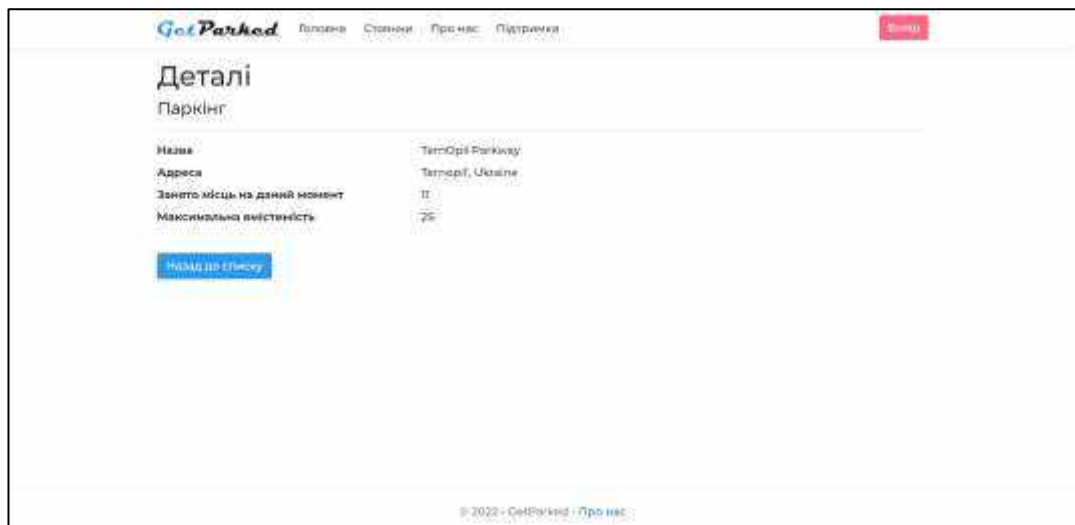


Рисунок 3.17 - Вигляд UI дизайну сторінки перегляду деталей стоянки

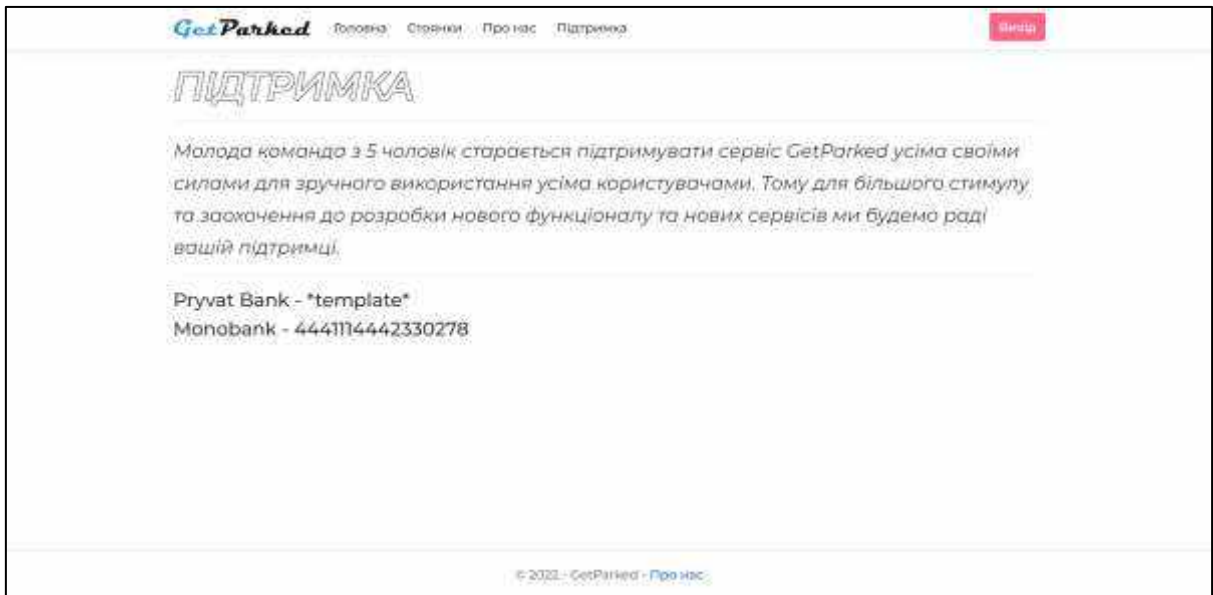


Рисунок 3.18 - Вигляд UI дизайну сторінки «Підтримка»

					ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

ВИСНОВКИ

В бакалаврській роботі була розглянута дуже актуальна проблема пошуку вільних парковочних місць у великих містах. Автор пропонує рішення цієї проблеми у вигляді розробки веб-застосунку для дистанційного інтелектуального виявлення вільних паркомісць. Використання інтелектуальних методів, таких як машинне навчання та обробка зображень, дозволяє створити надійний інструмент для виявлення вільних парковочних місць. Розроблений веб-застосунок може значно зменшити час, витрачений на пошук парковки, та зменшити навантаження на дорожні мережі.

Для досягнення поставленої мети, автор роботи провів аналіз існуючих рішень та технологій в галузі паркування, провів опитування серед користувачів та водіїв, що регулярно користуються послугами паркування. Також було розроблено та протестовано алгоритми для виявлення вільних паркомісць за допомогою машинного навчання та підкріпленням методом градієнтного спуску. В результаті був розроблений веб-інтерфейс застосунку для дистанційного інтелектуального виявлення вільного паркомісця.

Отже, бакалаврська робота має велике практичне значення для міст-мільйонників та їх мешканців, а також для компаній, що займаються послугами доставки та таксі, та міських влад. Розроблений веб-застосунок може покращити ефективність паркування та зменшити трафік на дорогах, що призведе до зменшення заторів та покращення якості життя мешканців міст. Також, автор роботи відзначає можливість подальшого розвитку розробленого веб-застосунку. Наприклад, можливість додавання додаткової функціональності, такої як інформація про наявність зарядних станцій для електромобілів або інформація про зупинки громадського транспорту. Також, автор зазначає, що розробка мобільного додатку для цієї системи може додатково збільшити його корисність та зручність для користувачів.

Важливим етапом розробки веб-застосунку було моделювання його функціональної та структурної складових. Це дозволило автору роботи

					ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

розробити веб-інтерфейс, який є зручним та інтуїтивно зрозумілим для користувачів. Додатково, розроблений алгоритм дистанційного виявлення вільних паркомісць на основі глибокого навчання з підкріпленням методом градієнтного спуску є надійним та точним, що є важливим чинником для коректної роботи веб-застосунку.

Однак, автором роботи було відзначено також деякі обмеження розробленої системи, зокрема, залежність від наявності відеокамер та підключення до відеопотоку. Також, низька якість зображень або незадовільне освітлення можуть впливати на точність роботи системи. Однак, автор зазначає, що ці обмеження є загальними для будь-якої системи виявлення вільних паркомісць та не є критичними для загальної ефективності веб-застосунку.

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Manville, M. (2014). Improving cities through parking policy. JOURNEYS.
2. Todd, L. (2006). Parking management: Innovative solutions to vehicle parking problems. Planetizen. Retrieved from <https://www.planetizen.com/node/21568>
3. Youssef, Kh., & Megahed, S. (2010). The perplexity of parking requirements: Standardization versus customization. In 8th International Architectural Conference, Architecture & Built Environment – Contemporary Issues. Assiut University.
4. Atif, Y., Ding, J., Jeusfeld, M.A. (2016). Internet of things approach to cloud-based smart car parking. Procedia Comput Sci, 98, 193-198.
5. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M. (2014). Internet of things for smart cities. IEEE Internet Thing. J., 1(1), 22-32.
6. Teodorovic, D., Lucic, P. (2006). Intelligent parking systems. Eur. J. Oper. Res., 175(3), 1666-1681.
7. SFpark. (n.d.). Retrieved from <http://sfpark.org>
8. Takizawa, H., Yamada, K., Ito, T. (2004). Vehicles detection using sensor fusion. Intelligent Vehicles Symposium, IEEE, pp. 238-243.
9. Funck, S., Mohler, N., Oertel, W. (2004). Determining car-park occupancy from single images. Intelligent Vehicles Symposium, IEEE, pp. 325-328.
10. Bong, D., Ting, K., Lai, K. (2008). Integrated approach in the design of car park occupancy information system (COINS). IAENG Int. J. Comput. Sci., 35(1), 7-14.
11. Wolff, J., Heuer, T., Gao, H., Weinmann, M., Voit, S., Hartmann, U. (2006). Parking monitor system based on magnetic field sensor. In 2006 IEEE Intelligent Transportation Systems Conference, IEEE, pp. 1275-1279.

					ДП.КН.8091455.069.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

12. Pala, Z., Inanc, N. (2007). Smart parking applications using RFID technology. 2007 1st Annual RFID Eurasia, IEEE, pp. 1-3.
13. Jian, M.S., Yang, K.S., Lee, C.L. (2008). Modular RFID parking management system based on existed gate system integration. WSEAS Trans. Syst., 7, 706-716.
14. Pham, T.N., Tsai, M.F., Nguyen, D.B., Dow, C.R., Deng, D.J. (2015). A cloud-based smart-parking system based on Internet-of-Things technologies. IEEE Access, 3, 1581-1591.
15. Geng, Y., Cassandras, C.G. (2013). New “smart parking” system based on resource allocation and reservations. IEEE Trans. intell. Transport. Syst., 14(3), 1129-1139.
16. Gallo, M., D'Acierno, L., Montella, B. (2011). A multilayer model to simulate cruising for parking in urban areas. Transp. Policy, 18(5 September), 735-744.
17. Benenson, I., Martens, K., Birr, S. (2008). Parkagent: an agent-based model of parking in the city. Comput. Environ. Urban Syst., 32(6 November), 431-439
18. Polak, J., & Axhausen, K. W. (1989). Clamp: a macroscopic simulation model for parking policy analysis. Paper presented at the 68th Annual Meeting Transportation Research Board, Washington DC.
19. Young, W. (1991). Parksim 1.1 User's Manual. Monash University: Department of Civil Engineering.
20. Rodier, C. J., & Shaheen, S. A. (2010). Transit-based smart parking: an evaluation of the San Francisco bay area field test. Transportation Research Part C: Emerging Technologies, 18(2), 225-233. <https://doi.org/10.1016/j.trc.2009.07.004>
21. Hodel, T. B., & Cong, S. (2003). Parking space optimization services, a uniformed web application architecture. In Proceedings of ITS World Congress (pp. 16-20).

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

22. Kumar, I., Manuja, P., Soni, Y., & Yadav, N. S. (2020). An integrated approach toward smart parking implementation for smart cities in India. In *Advances in Data and Information Sciences* (pp. 343-349). Springer.

23. Wang, H., & He, W. (2011). A reservation-based smart parking system. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 690-695). IEEE.

24. Nath, A., Konwar, H. N., Kumar, K., & Islam, M. R. (2020). Technology enabled smart efficiency parking system (TESEPS). In *Trends in Communication, Cloud, and Big Data* (pp. 141-150). Springer.

25. da Cruz, M. A., Rodrigues, J. J., Gomes, G. F., Almeida, P., Rabelo, R. A., Kumar, N., & Mumtaz, S. (2020). An IoT-based solution for smart parking. In *Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019)* (pp. 213-224). Springer.

26. Sarangi, M., Mohapatra, S., Tirunagiri, S. V., Das, S. K., & Babu, K. S. (2020). IoT aware automatic smart parking system for smart city. In *Cognitive Informatics and Soft Computing* (pp. 469-481). Springer.

27. Arena Simulation. (n.d.). Retrieved from <https://www.arenasimulation.com/>

28. Tanwar, S., Tyagi, S., Kumar, N., & Obaidat, M. S. (2018). LA-MHR: learning automata based multilevel heterogeneous routing for opportunistic shared spectrum access to enhance lifetime of WSN. *IEEE Systems Journal*, 13(1), 313-323.

29. Ren, J., Wu, G., Su, X., Cui, G., Xia, F., & Obaidat, M. S. (2016). Learning automata-based data aggregation tree construction framework for cyber-physical systems. *IEEE Systems Journal*, 12(2), 1467-1479.

30. Krishna, P.V., Misra, S., Saritha, V., Raju, D.N., & Obaidat, M.S. (2017). An efficient learning automata based task offloading in mobile cloud computing environments. In *2017 IEEE international conference on communications (ICC)* (pp. 1-6). IEEE.

31. Krishna, P.V., Misra, S., Saritha, V., Nagaraju, D., & Obaidat, M.S. (2016). Learning automata based decision making algorithm for task offloading in mobile cloud. In 2016 International Conference on Computer, Information and Telecommunication Systems, CITS (pp. 1-6). IEEE.

32. Mostafaei, H., & Obaidat, M.S. (2018). Learning automaton-based self-protection algorithm for wireless sensor networks. IET Networks, 7(5), 353-361.

33. Saritha, V., Krishna, P.V., Misra, S., & Obaidat, M.S. (2016). Learning automata-based channel reservation scheme to enhance QoS in vehicular adhoc networks. In 2016 IEEE Global Communications Conference, GLOBECOM (pp. 1-6). IEEE.

34. Saritha, V., Krishna, P.V., Misra, S., & Obaidat, M.S. (2017). Learning automata based optimized multipath routing using leapfrog algorithm for VANETs. In 2017 IEEE International Conference on Communications (ICC) (pp. 1-5). IEEE.

35. Trivedi, H., Tanwar, S., & Thakkar, P. (2018). Software defined network-based vehicular adhoc networks for intelligent transportation system: recent advances and future challenges. In Future Trends in Network Communication and Technology (pp. 325-337). Springer.

36. Chauhan, V., Patel, M., Tanwar, S., Tyagi, S., & Kumar, N. (2020). IoT enabled real-time urban transport management system. Computers & Electrical Engineering, 86, 106746.

37. Tyagi, S., Tanwar, S., & Kumar, N. (2015). Learning automata-based coverage oriented clustering in HWSNs. In 2015 Second International Conference on Advances in Computing and Communication Engineering (pp. 78-83). IEEE.

38. Chauhan, K., Jani, S., Thakkar, D., Dave, R., Bhatia, J., Tanwar, S., & Obaidat, M.S. (2020). Automated machine learning: the new wave of machine learning. In 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 205-212). IEEE.

39. Patel, D., Narmawala, Z., Tanwar, S., & Singh, P.K. (2019). A systematic review on scheduling public transport using IoT as tool. In Smart Innovations in Communication and Computational Sciences (pp. 39-48). Springer.

40. Oza, J., Narmawala, Z., Tanwar, S., & Singh, P.K. (2017). Public transport tracking and its issues. *International Journal of Computer Science and Engineering*, 5(11).

41. Mewada, A., Tanwar, S., & Narmawala, Z. (2018). Comparison and evaluation of real time reservation technologies in the intelligent public transport system. In *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 800-805). IEEE.

42. Tanwar, S., Tyagi, S., Kumar, N., & Obaidat, M. S. (2015). Learning automata-based coverage oriented clustering in HWSNs. In *2015 Second International Conference on Advances in Computing and Communication Engineering* (pp. 78-83). IEEE.

43. Chauhan, K., Jani, S., Thakkar, D., Dave, R., Bhatia, J., Tanwar, S., & Obaidat, M. S. (2020). Automated machine learning: the new wave of machine learning. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 205-212). IEEE.

44. Patel, D., Narmawala, Z., Tanwar, S., & Singh, P. K. (2019). A systematic review on scheduling public transport using IoT as tool. In *Smart Innovations in Communication and Computational Sciences* (pp. 39-48). Springer.

45. Oza, J., Narmawala, Z., Tanwar, S., & Singh, P. K. (2017). Public transport tracking and its issues. *International Journal of Computer Science and Engineering*, 5(11).

46. Mewada, A., Tanwar, S., & Narmawala, Z. (2018). Comparison and evaluation of real time reservation technologies in the intelligent public transport system. In *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 800-805). IEEE.

47. Tanwar, S., Tyagi, S., & Kumar, N. (2018). Traffic engineering in smart cities using software defined networking. In *Handbook of Research on Machine Learning Innovations and Trends* (pp. 416-442). IGI Global.

48. Chauhan, K., Narmawala, Z., & Tanwar, S. (2021). Review on vehicle detection and tracking techniques in intelligent transportation system. *Journal of Ambient Intelligence and Humanized Computing*, 1-16.

49. Tanwar, S., Tyagi, S., & Kumar, N. (2020). Automated machine learning for big data: State-of-the-art and future research directions. *Journal of Big Data*, 7(1), 1-28.

50. Narmawala, Z., Mewada, A., & Tanwar, S. (2019). An empirical study on real time location based services for public transport. In *Smart Innovations in Communication and Computational Sciences* (pp. 49-59). Springer.

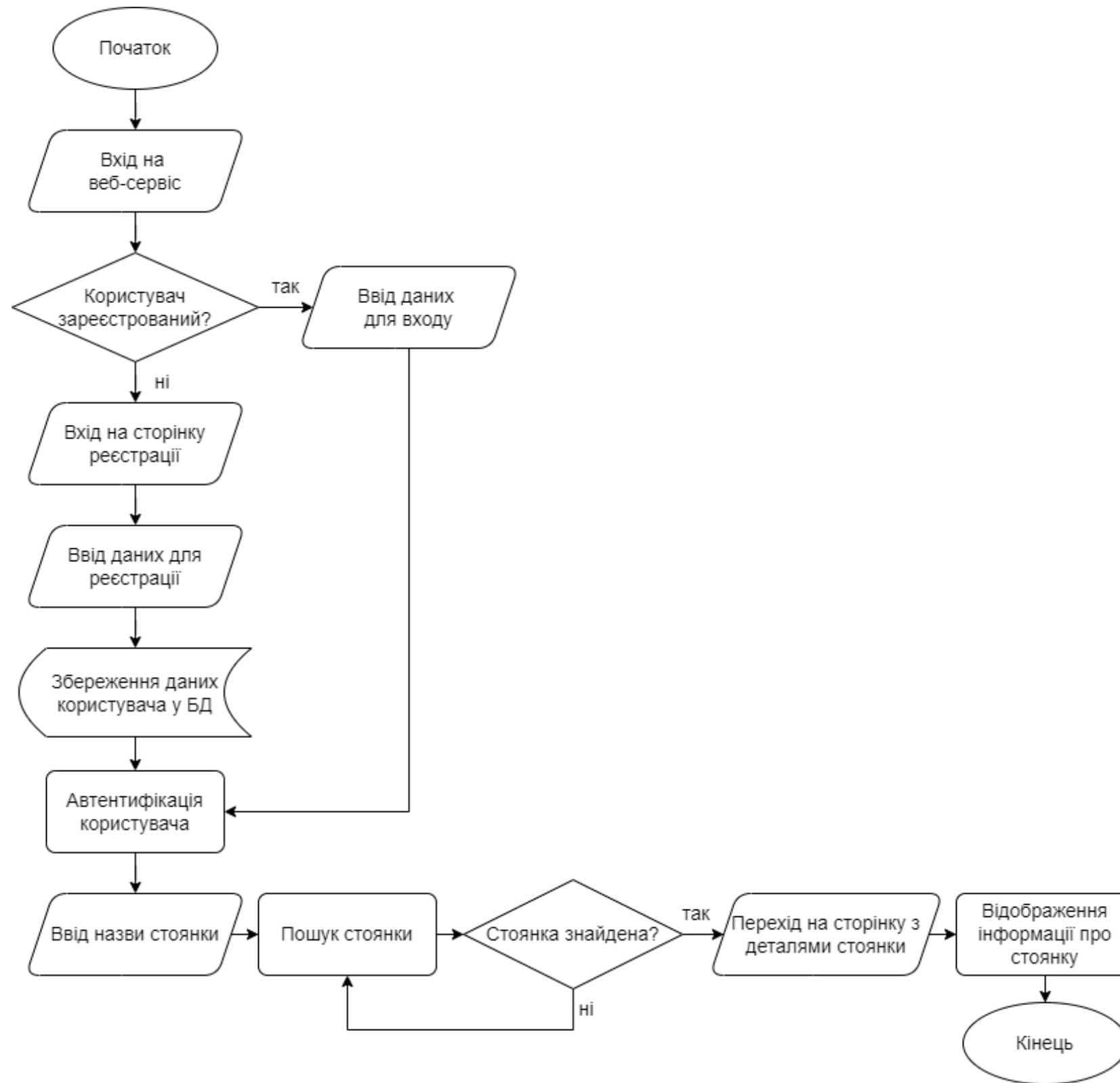
51. Lipianina-Honcharenko, K., Savchyshyn, R., Sachenko, A., Chaban, A., Kit, I., & Lendiuk, T. (2022). Concept of the Intelligent Guide with AR Support.

52. Komar, M., Savenko, O., Sachenko, A., Lendiuk, T., Lipianina-Honcharenko, K., Hladiy, G., & Vasylykiv, N. (2022). Evaluation the Efficiency of Information Technology of Big Data Intelligence Analysis and Processing.

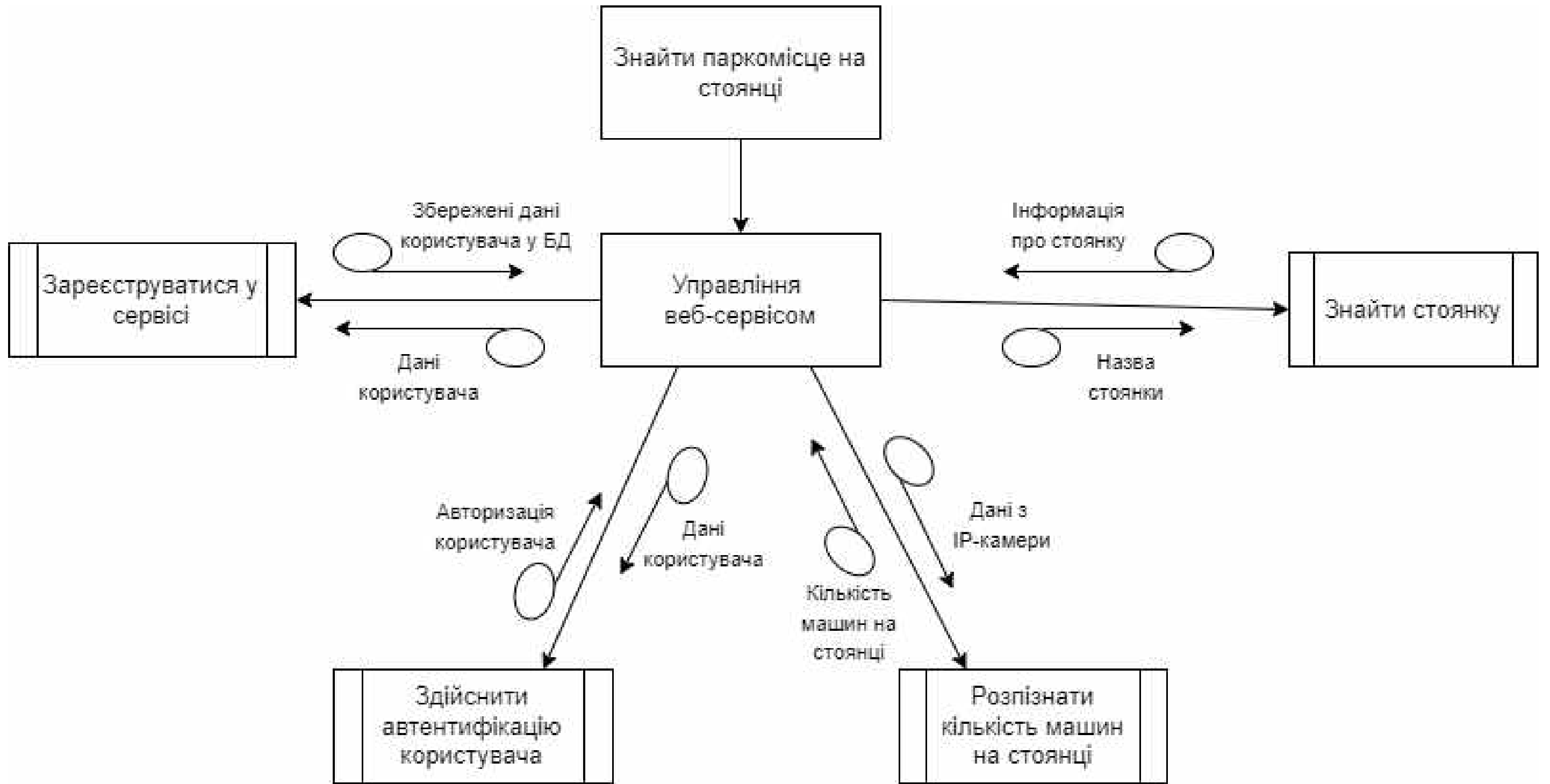
53. Kit, I., Zahorodnia, D., Lipyanina, H., Dorosh, V., Bykovyy, P., Kochan, V., ... & Grzeszczyk, K. (2020). Improved canny's method for laser scribes contour selection in solar cells. In *CEUR Workshop Proceedings* (pp. 395-405).

54. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. – Тернопіль: ЗУНУ, 2021. – 56 с.

					<i>ДП.КН.8091455.069.ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69



					ДП.КН. 8091455.001 А1			
Змн.	Арк.	№ докум.	Підпис	Дата	Функціональна схема організації роботи користувача для пошуку стоянки	Літера	Маса	Масштаб
Розроб.		Лучка С.						
Перевір.		Ліп'яніна-Гончаренко Х.В.						
Консультант						Аркуш 1	Аркушів 1	
Т. Контр.						ЗУНУ.ФКІТ.КН-41		
Н. Контр.		Ліп'яніна-Гончаренко Х.В.						
Затверд.		Комар М.П.						



					ДП.КН. 8091455.001 А1			
Змн.	Арк.	№ докум.	Підпис	Дата	Структурна карта Константайна модулів пошуку паркомісця на стоянці	Літера	Маса	Масштаб
Розроб.		Лучка С.						
Перевір.		Ліп'яніна-Гончаренко ХВ.						
Консультант								
Т. Контр.								
Н. Контр.		Ліп'яніна-Гончаренко ХВ.						
Затверд.		Комар М.П.				Аркуш 1	Аркушів 1	
						ЗУНУ.ФКІТ.КН-41		