

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота
на здобуття освітнього рівня магістра**

за спеціальністю 121 Інженерія програмного забезпечення
на тему:

**ДОСЛІДЖЕННЯ АЛГОРИТМІВ АНАЛІЗУ ЗОБРАЖЕНЬ НА
ПРИКЛАДІ ЗАДАЧІ ПОШУКУ МІСЦЯ НА ПАРКОВЦІ**

Виконав студент 2-го курсу магістратури
Максим ДОМАЛЬЧУК

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Максим ВЕРЕС

(підпис)

Засвідчую, що в цій роботі немає запозичень
з праць інших авторів без відповідних посилань.

Студент

(підпис)

Роботу розглянуто й допущено до захисту на
засіданні кафедри інтелектуальних програмних
систем

« 10 » травня 2023 р.,

протокол № 9

Завідувач кафедри

Олександр ПРОВОТАР

(підпис)

РЕФЕРАТ

Обсяг роботи 59 сторінок, 22 ілюстрації, 6 таблиць та 31 джерело посилань.

АНАЛІЗ ЗОБРАЖЕНЬ, НЕЙРОННА МЕРЕЖА, Google Cloud Vision, Python, IOS додаток.

Об'єктом розроблення є додаток для пошуку вільного паркувального місця за допомогою аналізу зображення парковки з камер з використанням нейронної мережі.

Метою роботи є розробка додатку, що вирішує задачу аналізу зображень для актуальної проблеми з використанням нейронної мережі.

Методи розроблення: програмування мовою Python, Swift, сервіси Google Cloud Vision та AWS EC2.

Результатами роботи є система для моніторингу стану паркувальних місць на основі зображень, проаналізовано різні підходи та можливості вирішення цієї задачі.

Програмний продукт, отриманий в результаті виконання даної роботи може бути використаний в подальшому як додаток для особистого використання або як основа для автоматизації комерційної парковки.

ЗМІСТ

ВСТУП	4
1. ОГЛЯД ЛІТЕРАТУРИ	6
1.1. Поняття нейромережі.	6
1.2. Класифікація нейронних мереж.	8
1.3. Принципи, за якими навчають нейронні мережі.	15
1.4. Аналіз існуючих технологій.	19
2. МЕТОДОЛОГІЯ	23
2.1. Збір та підготовка даних для навчання нейромережі.	23
2.2. Вибір та налаштування алгоритму для навчання нейромережі.	29
2.3. Вибір моделі та аналіз вимог до системи парковки на основі штучної нейронної мережі.	32
3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ	37
3.1. Архітектура системи для прокладання візуалізації та візуалізації маршрутів на основі технології доповненої реальності.	37
3.2. Опис розробленої системи знаходження вільних місць парковки.	44
3.3. Порівняння результатів роботи розробленої системи з результатами інших досліджень.	49
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ ДЛЯ РОЗУМНОЇ ПАРКОВКИ НА ОСНОВІ ТЕХНОЛОГІЇ КОМП'ЮТЕРНОГО ЗОРУ	57

ВСТУП

Зростаюча кількість автомобілів у сучасному світі зробила проблему пошуку вільного місця для паркування все більш актуальною. Без відповідної системи пошуку вільного місця для паркування водії можуть витратити значну кількість часу та нервів. Однак з появою технологій машинного навчання, розпізнавання зображень за допомогою нейронних мереж стало ефективним методом для вирішення цієї проблеми.

Застосування нейронних мереж для розпізнавання зображень дозволяє комп'ютерам з високою точністю відрізнити та класифікувати різні об'єкти на зображеннях. У випадку пошуку вільного місця для паркування нейронні мережі можуть бути навчені розпізнавати вільні та зайняті парковочні місця на зображеннях, отриманих з камер, розміщених на парковці. Після цього система може порівняти отримані дані з базою даних та повідомити водіїв про вільні парковочні місця в режимі реального часу.

Отже, використання нейронних мереж для розпізнавання зображень є потужним інструментом для вирішення проблеми пошуку вільного місця для паркування, що може сприяти ефективному використанню міського простору та зменшенню навантаження на водіїв.

Актуальність дослідження з розпізнавання зображень за допомогою нейронних мереж та пошуку вільного місця для паркування полягає в тому, що це важлива проблема для багатьох міст та населених пунктів. Не завжди можливо знайти вільне парковочне місце, що часто призводить до заторів на дорогах.

Головною метою дослідження є розробка системи зображення парковки та пошуку вільних місць за допомогою нейронних мереж. Завданням дослідження є створення програмного забезпечення, яке може обробляти зображення на вході та визначати, які парковочні місця вільні, а які зайняті.

Основною метою дослідження є скорочення часу, витраченого на пошук вільного місця для паркування та зменшення заторів на дорогах. Система також може бути корисною для водіїв з обмеженими можливостями, які потребують спеціальних місць для паркування. Дослідження може бути корисним для власників парковок, оскільки вони можуть відстежувати кількість вільних місць та контролювати плату за паркування. Загалом, розпізнавання зображень за допомогою нейронних мереж - це перспективна технологія, яка може допомогти вирішити проблему пошуку вільних місць для паркування та покращити урбаністичне середовище.

РОЗДІЛ 1. ОГЛЯД ЛІТЕРАТУРИ

1.1. Поняття нейромережі.

Нейронні мережі - це системи обчислень, які мають аналогії зі спрощеними біологічними системами людського мозку. Людський мозок обробляє інформацію за допомогою спеціальних біологічних елементів - нейронів. Біологічні нейронні мережі складаються з сукупності таких нейронів, які з'єднані між собою в центральній нервовій системі та гангліях. Рисунок 1.1 зображує будову біологічного нейрона.

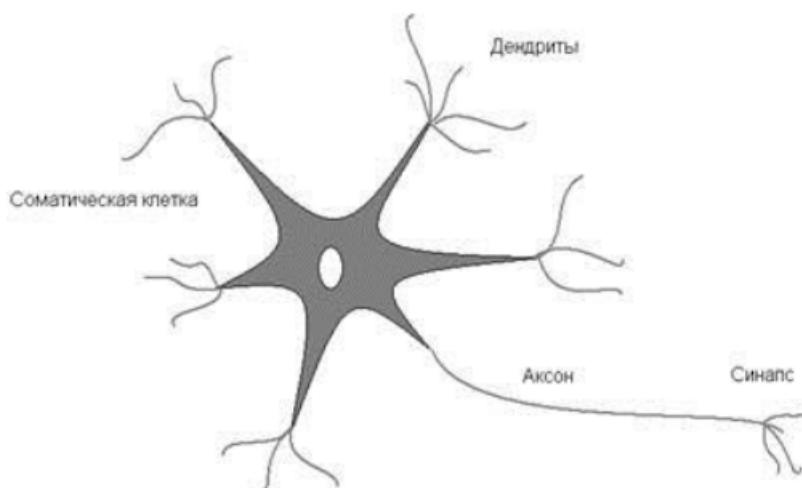


Рисунок 1.1 – Біологічний нейрон.

Біологічний нейрон приймає вхідний сигнал від інших нейронів або джерел інформації через дендрит, обробляє його в тілі, сомі, і передає вихідний сигнал через аксон до синапсу, який може бути з'єднаний з дендритами наступного нейрона. Один біологічний нейрон може бути з'єднаний з дендритами 20 тисяч інших нейронів, які передають йому вхідний сигнал, але на виході зазвичай присутній лише один аксон.

Аналогічно біологічному, штучний нейрон здатний приймати на вхід сигнал від багатьох нейронів, хоча їхня кількість зазвичай не досягає десятків тисяч. Будова штучного нейрона зображена на рисунку 1.2.

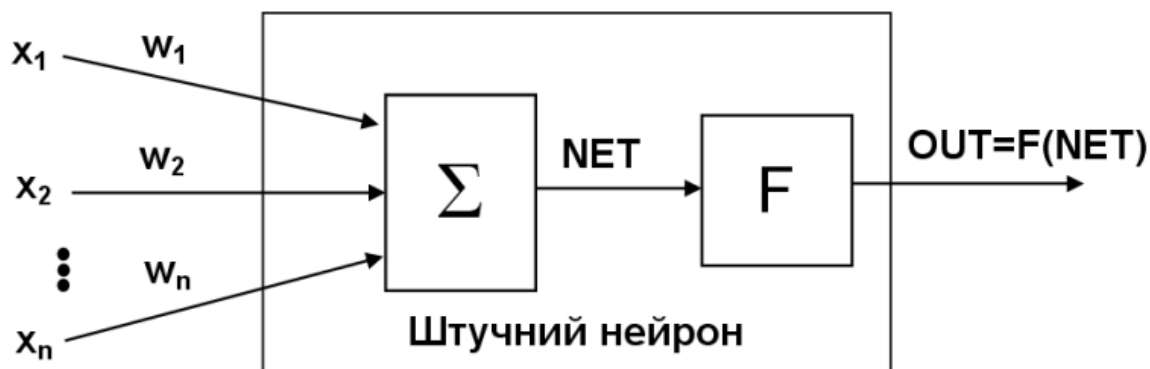


Рисунок 1.2 – Будова штучного нейрона.

Штучний нейрон має вагові коефіцієнти на входах, які визначають значимість зв'язків з іншими нейронами - менш вагомі зв'язки мають менші коефіцієнти ваги, а більш значимі зв'язки мають більші коефіцієнти ваги. Обробка інформації здійснюється за допомогою суматора та функції активації. Суматор перетворює виважені вхідні сигнали за формулою:

$$S = \sum w_i x_i \quad (1.1)$$

Після цього результат обробляється функцією активації за певною формулою:

$$Y = f(S) \quad (1.2)$$

Штучні нейрони в нейронних мережах зазвичай розташовані в шарах, в залежності від їх функції активації, що визначає, як буде оброблений вхідний сигнал. Кожен штучний нейрон має певну синоптичну вагу на своїх входах, що визначає важливість зв'язку. Сигнали поступово передаються від вхідних шарів до вихідних, пройшовши через приховані шари, якщо такі є. Зазвичай кожен нейрон вихідного шару та наступних шарів з'єднаний з кожним нейроном наступного шару, за винятком вихідного шару. Схему простої штучної нейронної мережі з одним прихованим шаром нейронів можна побачити на рисунку 1.3.

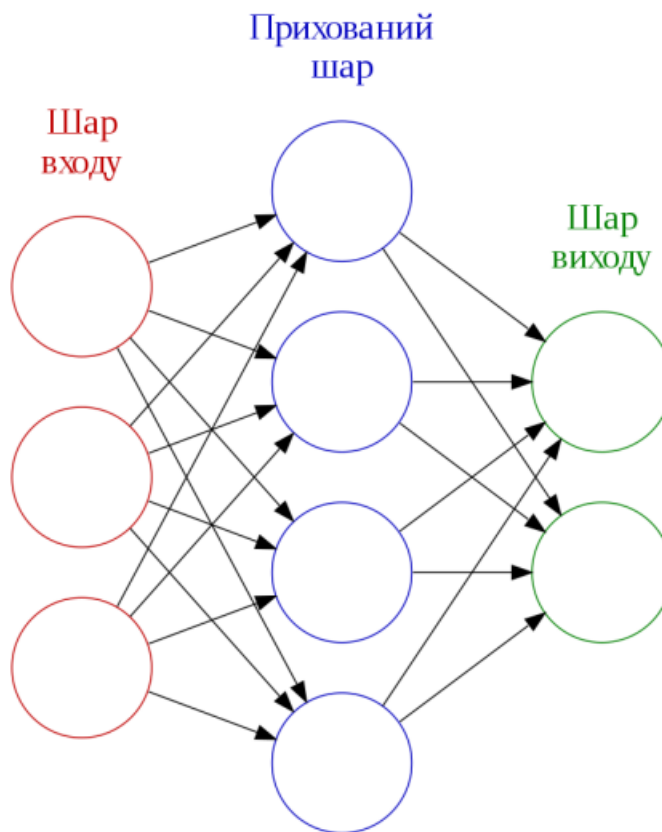


Рисунок 1.3 – Схема простої нейронної мережі із одним прихованим шаром.

1.2. Класифікація нейронних мереж.

Нейронні мережі мають різноманітні архітектури, які використовуються для різних завдань, залежно від того, яка архітектура є найбільш підходящою для конкретного завдання. Нейромережу можна представити у вигляді графіка зі зваженими зв'язками, де кожен нейрон представлений у вигляді вузла. Залежно від архітектури зв'язків, нейронні мережі можна розділити на два класи (рис.1.4): мережі з прямим поширенням, де графіки не мають циклів, та рекурентні мережі, які є зворотними мережами.

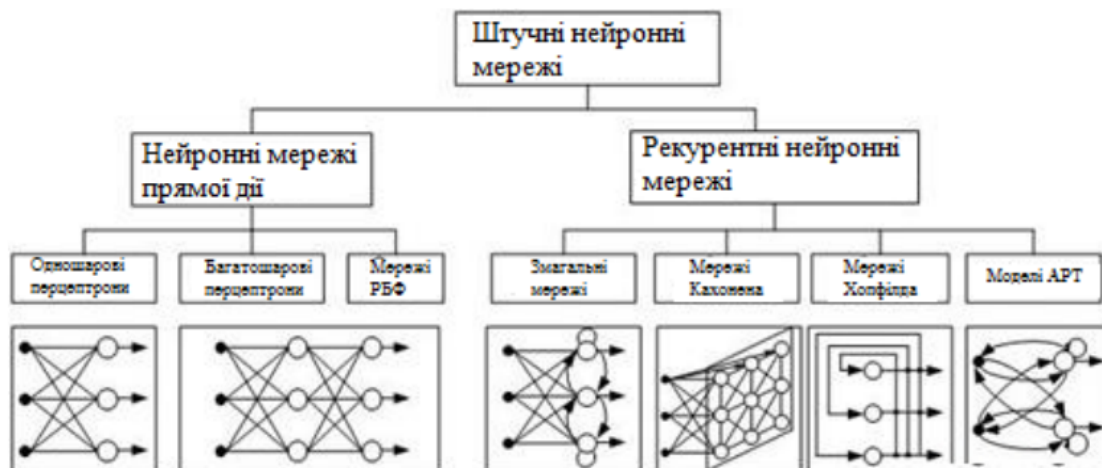


Рисунок 1.4 – Класифікація нейронних мереж за архітектурою.

Нейронні мережі мають різноманітність архітектур, які використовуються для виконання завдань, що підходять найкраще для конкретної архітектури. Наприклад, мережа РБФ використовує радіальні базисні функції як функції активації.

Нейронні мережі також відрізняються за:

- структурою мережі;
- особливостями моделі нейрона;
- особливостями навчання мережі.

За структурою нейронні мережі поділяються рис. 1.5. на:

- Неповно зв'язні і повно зв'язні;
- З випадковими і регулярними зв'язками;
- З симетричними і несиметричними зв'язками.



Рисунок 1.5 – Класифікація нейромереж за структурою.

Нейронні мережі можуть бути поділені за топологією на три основні типи:

1. Неповно зв'язні мережі, які поділяються на одношарові та багатшарові з прямими, перехресними та зворотними зв'язками. У мережах з прямими зв'язками, нейрони j -го шару на входах можуть з'єднуватися лише з нейронами i -го типу, де $j > i$, тобто з нейронами нижчих шарів. У мережах з перехресними зв'язками дозволені з'єднання всередині одного шару, тобто вищевказана нерівність замінюється на $j \geq i$. У нейромережах із зворотніми зв'язками використовуються i -ті зв'язки j -го шару з входами до i -го при $j < i$.
2. Повно зв'язні мережі, в яких кожен нейрон підключений до всіх інших нейронів мережі. Такі мережі часто використовуються для задач класифікації та розпізнавання зразків.
3. Самоорганізуючі мережі, які використовуються для кластеризації та карти Кохонена. У таких мережах кожен нейрон пов'язаний зі своїми найближчими сусідами, що утворює регулярну геометричну структуру.

За топологією можна виділити три основні типи нейронних мереж:

- повно зв'язні (рис. 1.6.а);
- багатшарові (рис. 1.6.б);
- слабо зв'язані (рис. 1.6.в)

У повно зв'язних нейронних мережах кожен нейрон передає свій вихід іншим нейронам, включаючи себе, а вхідні сигнали передаються на всі нейрони. Після кількох циклів роботи мережі вихідні сигнали можуть бути отримані з усіх або частини нейронів. В мережах з багатьма шарами нейрони групуються в окремі шари, кожен з яких містить набір нейронів з унікальними вхідними сигналами. Кількість нейронів в кожному шарі може бути різною, і не залежить від кількості нейронів в інших шарах.

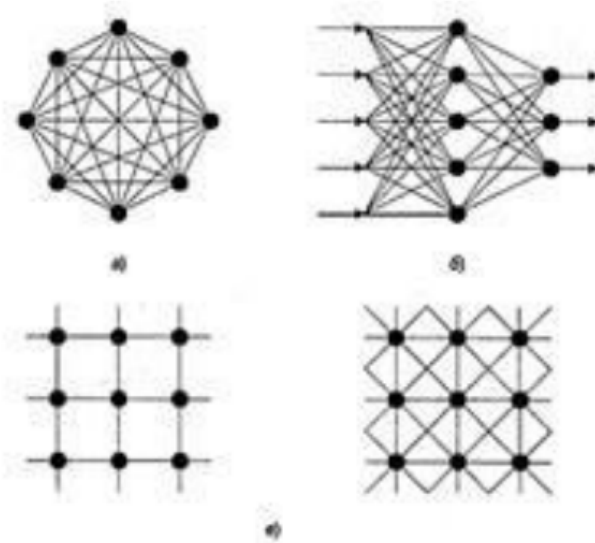


Рисунок 1.6. – Архітектури нейронних мереж: а – повно зв'язна мережа, б – багатошарова мережа з послідовними зв'язками, в – слабо зв'язна мережа.

В загальному випадку, багатошарова нейронна мережа складається з Q шарів, які пронумеровані зліва направо. Зовнішні вхідні сигнали передаються на вхідні нейрони вхідного шару, а виходи мережі є вихідними сигналами останнього шару. Крім вхідного і вихідного шарів, багатошарової нейронної мережі може містити один або кілька прихованих шарів. З'єднання виходів нейронів шару q зі входами нейронів наступного шару ($q + 1$) називають послідовними.

У світі багатошарових нейронних мереж існує кілька типів, серед яких можна виділити:

1) Монотонні мережі - це спеціальний випадок багаторівневих мереж з додатковими нейронами, де кожен шар, крім останнього, складається з двох блоків: збудливого та гальмуючого. Зв'язки між цими блоками можуть бути як збудливими, так і гальмуючими. Якщо між нейронами блоку А та нейронами блоку В існують тільки збудливі зв'язки, то будь-який вихідний сигнал блоку В є монотонно неспадаючою функцією для будь-якого вихідного сигналу блоку А. Якщо ж зв'язки між нейронами блоку А та нейронами блоку В є гальмуючими, то будь-який

вихідний сигнал блоку В є незростаючою функцією для будь-якого вихідного сигналу блоку А. У монотонних мережах для нейронів необхідна монотонна залежність вихідного сигналу від параметрів вхідних сигналів.

2) Мережі без зворотних зв'язків - це мережі, у яких нейрони вхідного шару отримують вхідні сигнали, перетворюють їх та передають в нейрони першого прихованого шару. Процес перетворення сигналів повторюється на наступних прихованих шарах, поки сигнали не досягнуть вихідного шару, який видає сигнали для інтерпретатора та користувача.

Зазвичай вихідний сигнал q - го шару передається на вхід всіх нейронів $(q + 1)$ -го шару, якщо не вказано інше. Але існує можливість з'єднання шару q з довільним шаром $(q + p)$. Багат шарові мережі без зворотного зв'язку можуть бути повнозв'язними, тобто кожен вихідний нейрон q -го шару пов'язаний з входами всіх нейронів $(q + 1)$ -го шару, або частково повно зв'язними.

Класичним прикладом повно зв'язних мереж є мережі прямого поширення. В цих мережах вхідні дані проходять через шари нейронів, що знаходяться між вхідними і вихідними шарами. Вихідні дані генеруються на виході останнього шару і передаються для інтерпретації або використання користувачем.

3) Мережі зі зворотними зв'язками. В таких мережах інформація з наступних шарів передається на попередні. Як приклад мереж із зворотними зв'язками на рис. 1.7. представлені частково-рекурентні мережі Елмана і Жордана.

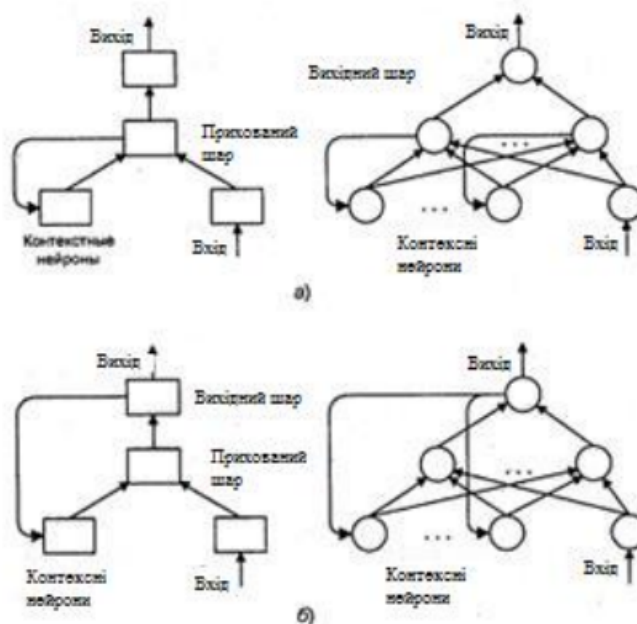


Рисунок 1.7. – Частково-рекурентні мережі: а – Елмана, б – Жордана.

У слабо зв'язаних нейромережах нейрони розташовані у вузлах прямокутної або гексагональної решітки, де кожен нейрон з'єднаний з чотирьома, шістьма або вісьмома своїми сусідами. В залежності від типу сигналів, якими вони операціонують на входах та виходах, слабо зв'язані нейромережі можуть бути аналоговими або бінарними. У бінарних нейромережах вихід кожного нейрона може приймати значення логічного нуля або логічної одиниці, що відповідає його загальмуванню або збудженню.

Серед найбільш поширених моделей слабо зв'язаних нейромереж можна виділити:

- Модель Хопфілда;
- Машина Больцмана;
- Мережу Кохонена;
- Рекурентну нейромережу;
- Згорткову нейромережу;
- Багатошаровий та одношаровий перцептрон.

Кожна з цих моделей має свою структуру та особливості використання, які дозволяють вирішувати різноманітні задачі, такі як класифікація, кластеризація, прогнозування та розпізнавання зображень.

Прикладом класичної архітектури мереж прямого поширення може слугувати – повно зв'язні нейромережі прямого поширення, або FNN.

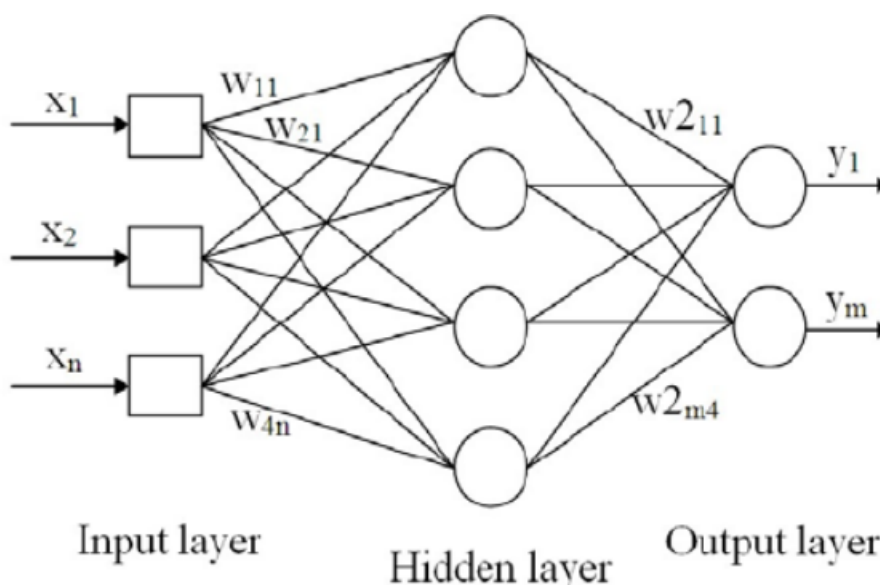


Рисунок 1.8. – Багатошаровий персептрон.

Персептрон, що був показаний на рис. 1.8, ще називають класичною нейронною мережею, яка є повністю зв'язаною, оскільки кожен нейрон пов'язаний з усіма нейронами попереднього шару. Ця мережа може успішно виконувати завдання класифікації. Проте, в нього є дві проблеми:

а) Перша проблема багатошарових нейронних мереж полягає у великій кількості параметрів, які потрібно оптимізувати. Наприклад, при обробці зображень $100 * 100$ пікселів в 3-шаровій мережі з кількістю нейронів на кожному шарі близько мільйона, загальна кількість параметрів може досягати мільйона. Це створює велику навантаження на оптимізацію та навчання мережі, та потребує великої кількості даних для навчання. Більш того, мережі з великою кількістю параметрів можуть мати тенденцію до перенавчання та повторного використання шумових даних.

б) Друга проблема - затухання градієнтів - виникає, коли мережа має багато шарів. Градієнти, які передаються від заднього до переднього шару під час зворотного поширення помилки, можуть ставати дуже малими в кінці мережі. Це може призвести до того, що нижчі шари мережі не будуть навчатися, оскільки градієнти будуть занадто малі для того, щоб змінити ваги нейронів. Ця проблема може бути подолана за допомогою різних методів, таких як використання інших функцій активації або нормалізація мережі.

1.3. Принципи, за якими навчають нейронні мережі.

Навчання нейромережі ШНМ – це процес налаштування параметрів мережі, що включає моделювання середовища, в яке вбудована модель. У цьому процесі використовують метод параметризації. Можна розрізнити два типи алгоритмів навчання мереж: з вчителем та без нього.

Перед тим, як почати використовувати нейромережу, її потрібно навчити, як це показано на рис. (див. рис. 1.9). Під час навчання з вчителем, мережі надається набір навчальних прикладів. Зразок надсилається на вхід мережі, потім обробляється в структурі нейронної мережі та обчислюється вихід мережі, який порівнюється з відповідним значенням цільового вектора, що представляє бажаний вихід мережі.

Далі, згідно з обраним алгоритмом, обчислюється похибка, та вагові коефіцієнти зв'язків в середині мережі змінюються. Послідовно подаються вектори навчального набору, обчислюються похибки та регулюються ваги для кожного вектора, поки похибка в навчальній матриці не досягне прийняттого рівня.

Під час навчання без вчителя, навчальний набір містить лише вхідні вектори. Алгоритм навчання коригує ваги мережі з метою отримання когерентних векторів виходу, тобто, що подання досить близьких вхідних векторів дає однакові результати.

Таким чином, сам процес навчання відрізняє статистичні властивості навчального набору і групує схожі вектори на класи. Подання вхідного вектора даного класу дасть певний вихідний вектор, але перед тим, як дізнатись, який вихід буде наданий цим вхідним векторним класом, неможливо передбачити.

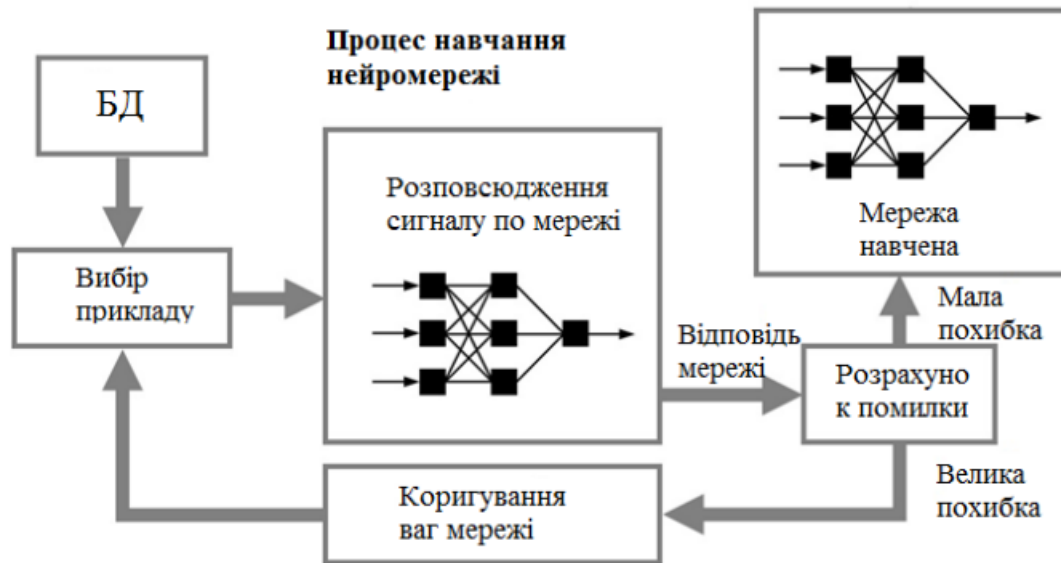


Рисунок 1.9. – Ілюстрація процесу навчання нейронної мережі.

Під час навчання нейронної мережі без вчителя методами сигналізації Гейбба та Ойя використовуються для зміни ваг зв'язків в середині мережі з метою отримання когерентних векторів виходу, тобто таких, що подання досить близьких вхідних векторів дає однакові результати.

Під час роботи нейронна мережа генерує вихідний сигнал Y , який реалізується за допомогою функції $Y = G(X)$, де X - вхідний сигнал, а G - функція, яка визначається синаптичним значенням ваг та зміщенням мережі, залежно від її архітектури. Під час процесу навчання ваги та зміщення мережі змінюються з метою досягнення бажаного вихідного сигналу. Результати навчання мережі слід перетворити на зрозумілу форму, обумовлену процесом навчання, що не є складним завданням, оскільки зв'язок між входом і виходом, встановлений мережею, зазвичай досить простий і очевидний.

Нехай розв'язанням задачі є функція $Y = F(X)$, задана параметрами вхідновихідних даних $(X^1, Y^1), (X^2, Y^2), \dots, (X^N, Y^N)$, для якого $Y^k = F(X^k) (k = 1, 2, \dots, N)$. Тренування складається з пошуку (синтезу) функції G , близької до F за сенсом узгодження функції помилки E див. рис. 1.10.

Якщо обрано багато прикладів навчання – пари $(X^k, Y^k) (k = 1, 2, \dots, N)$ та метод обчислення функції помилки E , то вивчення нейронної мережі перетворюється на багатовимірну оптимізацію задачі, яка має дуже велику розмірність, оскільки функція E може мати довільну форму навчання в загальному випадку і багатократну опуклу задачу про врегулювання часу.

Для вирішення цієї проблеми можна використовувати такі алгоритми:

1) Локальні алгоритми оптимізації для обчислення приватних похідних першого порядку:

- алгоритм градієнта (метод найшвидшого спуску),
- методи одновимірної та двовимірної оптимізації цільової функції в напрямку антиградієнта,
- метод спряженого градієнта,
- методи, що враховують напрямок антиградієнта в декількох кроках
- алгоритму.

2) Алгоритми локальної оптимізації для обчислення приватних похідних першого та другого порядку:

- метод Ньютона,
- методи оптимізації рідкісних гессіанських матриць,
- квазіньютонівські методи,
- метод Гаусса-Ньютона,
- метод Левенберга-Маркарда та інші.

3) Алгоритми стохастичної оптимізації:

- випадковий пошук,
- імітований відпал,
- метод Монте-Карло (цифровий метод статистичних випробувань);

4) Алгоритми глобальної оптимізації (проблеми глобальної оптимізації вирішуються шляхом сортування значень змінних, від яких залежить цільова функція).

Існують чотири основні правила навчання, пов'язані з пов'язаними мережевими архітектурами: виправлення помилок, правило Больцмана, правило Хебба та метод конкуренції.

Виправлення помилок.

Для кожного вхідного прикладу визначається очікуваний вихід (цільовий), який може не збігатися з фактичним виходом. Правило корекції полягає у використанні помилки - різниці між цільовим значенням та передбачуваним значенням для коригування ваг нейронів з метою зменшення неузгодженості. Тренування проводиться тільки в разі помилкового результату. Існує багато модифікацій цього правила навчання.

Правило Больцмана.

Правило Больцмана є стохастичним правилом навчання, яке базується на аналогії з термодинамічними принципами. При його виконанні ваги нейронів коригуються відповідно до необхідного розподілу ймовірностей. Вивчення правила Больцмана можна розглядати як окремий випадок корекції помилок, де помилка визначається як різниця кореляцій стану в двох режимах.

Правило Хебба.

Один з перших підходів до навчання без викладача у ШНМ - це правило Д. Хебба, яке у своєму нейрофізіологічному аспекті стверджує наступне: якщо аксон клітини А знаходиться на відстані, достатньо близькій до клітини В і постійно та періодично бере участь у її активації, то в одному

або обох нейронах відбуваються метаболічні зміни, що призводять до того, що ефективність нейрона А як збудника нейрона В підвищується.

Метод змагання

Алгоритм зворотного поширення помилок є одним із найпотужніших алгоритмів навчання ШНМ. Він відрізняється від правила Хебба тим, що вихідні нейрони змагаються між собою, і тільки вихідний нейрон з максимальним значенням зваженої суми є "переможцем", а виходи інших вихідних нейронів встановлюються в неактивний стан. На тренуванні змінюються лише ваги нейронів "переможець" у сенсі збільшення близькості до цього вхідного прикладу.

Основна ідея алгоритму зворотного поширення помилок полягає в тому, що зміна ваги синапсу враховує локальний градієнт функції помилки. Різниця між фактичною та правильною мережевою відповіддю, визначеною на вихідному шарі, поширюється у зворотному напрямку(рис. 1.10), у напрямку потоку сигналу.

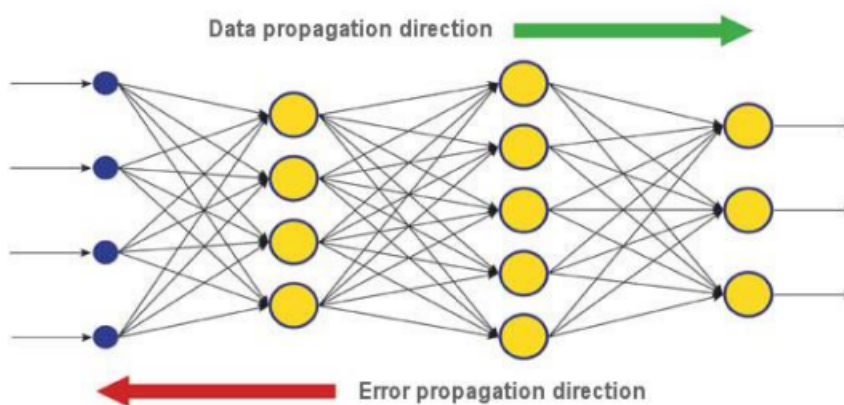


Рисунок 1.10. – Схема поширення даних і помилки в нейромережі при навчанні методом зворотного поширення помилки.

Під час навчання нейронної мережі за допомогою методу зворотного поширення помилок, ваги синапсів змінюються залежно від локального градієнту функції помилки, що дозволяє оцінити внесок кожної ваги у загальну помилку мережі. Однак, цей метод може привести до попадання в локальний мінімум, тому використовуються спеціальні прийоми для виходу з

нього. Якщо після декількох ітерацій алгоритм збігається до того ж рішення, можна зробити висновок про те, що знайдене рішення, швидше за все, є оптимальним.

1.4. Аналіз існуючих технологій.

Зі зростанням урбанізації та кількості автомобілів у великих містах, проблема паркування стала важливою як для практичних застосувань, так і з точки зору наукових досліджень. Для знаходження нових раціональних рішень, щодо паркування автомобілів у містах, було запропоновано концепцію розумного міста.

Ця проблема вивчалась у працях вчених з усього світу. Наприклад, Lookmuang та ін. розробили прототип з використанням вбудованого контролера, Raspberry Pi 3, ультразвукового датчика та камери для локалізації вільних місць для паркування. Подібний підхід запропонували також Vakula та інші та Grodi та інші, які обговорили переваги та недоліки декількох типів датчиків, таких як індукційне наближення, RFID, виявлення світла/дальності та камера. У обох випадках автори вирішили використовувати ультразвуковий датчик для визначення того, коли транспортний засіб був припаркований на місці паркування.

Ще одну розумну систему паркування, яка поєднує в собі різні технології IoT, такі як ультрависокочастотна RFID (UHF RFID), бездротова сенсорна мережа (WSN) і зв'язок ближнього поля (NFC), представили Maine та інші. Робота Patane та інші є прикладом системи виявлення вільних паркувальних місць, яка використовує систему зору. Аналогічний підхід до класифікації вільних місць для паркування був запропонований Amato та інші, де підготували колекцію з приблизно 150 000 зображень для класифікації вільних місць для паркування.

Незважаючи на існуючі дослідження у галузі інтелектуальних систем паркування, на сьогоднішній день не існує готового до використання рішення, яке б забезпечувало швидкий і простий доступ для користувачів з

низькими витратами на установку та обслуговування. Тому пошук методів розумного паркування є актуальною проблемою.

Існує кілька запропонованих систем, що використовуються для пошуку вільних місць для паркування. Одна з таких систем передбачає встановлення невеликих датчиків на кожному парковому місці, або розміщення камер поруч, які відстежують наявність вільних місць. Однак ці рішення потребують джерела живлення та частого обслуговування кожного датчика. Кожна з цих систем має свої переваги та недоліки, що залежать від сенсорної технології, що використовується для визначення наявності вільних місць для паркування.

Сьогодні існує кілька запропонованих технологій для розвитку системи інтелектуального паркування, яка забезпечує швидкий та легкий доступ для користувача з низькими витратами на встановлення та обслуговування.

а) Однією з технологій є ультразвуковий датчик, який використовується для виявлення руху, присутності, наближення або вимірювання відстані. Ця технологія широко застосовується в різних галузях, таких як вода та стічні води, видобуток корисних копалин, загальна промисловість, хімія та нафтохімія.

б) RFID є іншою технологією, яка широко використовується в промисловості, електронній комерції, кредитних картках, ID картках та інших галузях. RFID є ключовою технологією для розвитку рішень Інтернету речей, зокрема, для систем паркування. Ця технологія використовується для виявлення транспортних засобів.

в) Магнітний датчик є ще однією технологією, яка може бути корисною при розробці системи виявлення вільних місць для паркування. Він виявляє аномальні зміни магнітного поля, що дозволяє визначити, чи зайняте парковочне місце.

г) Камери можуть використовуватись як альтернатива датчикам для виявлення транспортних засобів на місцях для паркування. Одним з

основних недоліків датчиків є необхідність використання окремого датчика на кожне паркувальне місце, що може бути дорогим рішенням. Системи на основі зору можуть забезпечувати більш масштабне рішення, використовуючи камеру, розташовану в зоні паркування з широким оглядом, щоб визначити стан декількох паркувальних місць. Таблиця 1.1 порівнює різні методи та технології, які використовуються для розумних систем паркування.

Метод	Опис	Переваги	Недоліки
RFID	Заснована на використанні радіочастотного ідентифікатора	Далекий зв'язок, висока точність	Вартість технології, необхідність установки додаткового обладнання
Ультразвуковий датчик	Заснована на використанні відбиття ультразвукових хвиль	Дешева технологія, точність вимірювання відстані	Вплив акустичних перешкод, обмежений зону охоплення
Магнітний датчик	Заснована на зміні магнітного поля	Не чутливий до віддаленості, висока точність	Обмежена зона охоплення, можлива помилкова реакція на інші джерела магнітного поля
Камера	Заснована на використанні обробки відеоінформації	Можливість розпізнавання автомобілів та інших деталей, широкий охоплюваний діапазон	Вплив погодних умов, обмежена точність
ІоТ технології	Використання різноманітних сенсорів та засобів ІоТ	Гнучкість, можливість віддаленого контролю	Вартість технології, потреба у високій швидкості Інтернет-з'єднанн

			я
--	--	--	---

Таблиця 1.1. Порівняння різних методів технології обробки зображень місць парковки.

Загальний аналіз наведеної таблиці дозволяє зробити висновок, що метод розумного паркування з використанням камери є найбільш оптимальним в порівнянні з іншими розглянутими методами. Цей метод має ряд переваг, зокрема, він забезпечує більш точне виявлення вільних парковочних місць, масштабованість та більш низькі витрати на обладнання. Крім того, так як більшість паркувальних місць розташовані в громадських місцях, метод на основі камери повинен враховувати фактори приватності та безпеки. У цілому, метод розумного паркування з використанням камери є більш ефективним і перспективним в порівнянні з іншими методами.

РОЗДІЛ 2. МЕТОДОЛОГІЯ

2.1. Збір та підготовка даних для навчання нейромережі.

Сьогодні на ринку існує багато проектів розумних паркінгів, проте готові до використання приклади можна перерахувати на пальцях однієї руки, а інформація про економічний аспект їх реалізації, як правило, мінімальна. Варто зазначити, що при розробці таких інструментів найважливішу фінансову частину розробки несе програмне, а не апаратне забезпечення. Після того, як були розглянуті та порівняні різні методи та підходи до розумних парковок, можна зробити висновок, що метод розумного паркування з використанням камер зовнішнього спостереження є набагато ефективнішим за інші, враховуючи більшість факторів.

Методи, що базуються на технології комп'ютерного зору, складаються з дешевих камер, які охоплюють всю стоянку. Камери відеоспостереження, які використовуються для спостереження, також можна використовувати для виявлення наповненості. Зображення, зроблені з цих камер, згодом обробляються для отримання інформації про зайнятість. Однак є дві проблеми, які обмежують широке застосування методів, заснованих на баченнях.

Перша проблема полягає в низькій точності виявлення транспортних засобів на зображеннях за допомогою методів на основі комп'ютерного зору, порівняно з методами на основі підрахунку або сенсорів. Це може бути пов'язано з різноманітним зовнішнім виглядом транспортних засобів, такими як тіні, відображення та туман, а також з оклюзією іншими транспортними засобами або іншими об'єктами в лінії огляду, а також із спотворенням зображень через недостатній огляд камер.

Друга проблема полягає в розмежуванні паркувальних місць на зображеннях. Це розмежування не є необхідним для методів на основі лічильників, оскільки кількість місць для паркування фіксована, а для методів на основі датчиків кожне місце для паркування фізично відмежовується один

раз, щоб встановити датчик. Зона паркування може бути охоплена кількома камерами, а можливо, і сотнями камер для паркування на вулиці. Ручне маркування кожного паркувального місця є трудомістким завданням. Крім того, межі паркування можуть час від часу змінюватися. Інша пов'язана проблема виникає в районах, де місця для паркування не позначені, особливо в країнах, що розвиваються, таких як Україна.

В багатьох країнах, зокрема в Україні, проблема паркування полягає у тому, що місця для паркування не позначені. Це створює проблему в тому, як водії мають знайти де припаркуватися. Крім того, нерідко стається так, що транспортні засоби паркуються неправильно або незаконно, наприклад, коли припаркований автомобіль перешкоджає русу автобусів або розмітці між двома місцями. Тому важливо мати автоматичні способи виявлення місць для паркування, особливо в немаркованих зонах.

Для точного виявлення місць для паркування використовують надійні зображення та новітні методи комп'ютерного зору. Такі рішення є дуже бажаними для інтелектуального паркування, оскільки вони дозволяють визначати, де можна припаркуватися і де не можна. Оскільки більшість паркувальних місць знаходяться в громадських місцях, важливо враховувати фактори безпеки та приватності.

Отже, для досягнення поставленої мети дослідження необхідно виконати наступні завдання:

- Провести аналіз сучасних технологій розпізнавання зображень на основі штучних нейронних мереж.
- Вибрати оптимальну технологію для створення кібер-фізичної системи розумного паркування, яка буде базуватися на камері зовнішнього спостереження паркінгу університету.
- Розробити інформаційну модель, яка буде здатна виявляти паркувальні місця та ідентифікувати транспортні засоби.
- Провести оцінку розробленої моделі на наборі даних тестування.

При виконанні цих завдань необхідно враховувати фактори приватності та безпеки, оскільки більшість паркувальних місць розташовані в громадських місцях. Крім того, важливо враховувати особливості країни, в якій буде використовуватися система, зокрема в Україні, де паркування часто не позначене і може бути неправильно здійснене. Бажано використовувати надійні зображення для точного виявлення паркувальних місць, а також застосовувати новітні методи комп'ютерного зору для досягнення більш точних результатів.

Протягом десятиліть вивчається метод виявлення паркувальних місць на основі розпізнавання образів. Традиційні методи можна розділити на дві категорії: лінійні та на основі розмітки. Лінійні методи виявляють лінії на зображенні навколо огляду за допомогою різних алгоритмів, таких як детектор краю Канні, оператор Лапласа та каскад Хаара, і потім прогнозують параметри виявлених ліній за допомогою алгоритма підгонки ліній. Геометричні обмеження, розроблені вручну, застосовуються для фільтрації та визначення місця для паркування. Методи на основі розмітки спочатку знаходять точки розмітки на зображенні навколо огляду за допомогою детектора кутів Харріса або підсилюючого дерева рішень, а потім використовують техніку відповідності шаблону або комбінують виявлення ліній для визначення місця для паркування.

Традиційні методи виявлення паркувальних місць дають хороші результати, але вони чутливі до змін навколишнього середовища і не застосовуються до складного реального середовища. Останні роки були присвячені пошуку найкращих підходів для виявлення паркувальних місць і розпізнавання транспортних засобів, і у таблиці 2.1 представлені гідні дослідження в цій області.

Підхід	Опис	Переваги	Недоліки
Використання ліній	Виявлення паркувальних місць шляхом	Простота реалізації; висока точність для	Чутливість до зміни навколишнього середовища; не

	визначення ліній на зображенні	виявлення ліній на зображенні; можливість застосування на різних типах парковок	можливість визначення типу транспорту за формою паркуючого місця; недостатньої точності визначення місця для паркування; не можливість прогнозування місця для паркування при зміні огляду камери
Використання точок	Виявлення паркувальних місць шляхом визначення точок розмітки на зображенні	Простота реалізації; можливість використання на різних типах парковок; можливість визначення типу транспорту за формою паркуючого місця	Чутливість до зміни навколишнього середовища; недостатня точність визначення місця для паркування; не можливість прогнозування місця для паркування при зміні огляду камери
Використання ІНС	Виявлення паркувальних місць та позиції автомобілів з використанням інерціальних сенсорів	Незалежність від зовнішніх факторів; висока точність визначення позиції автомобіля; можливість використання на парковках будь-якого типу	Недостатня точність визначення місця.

Таблиця 2.1. Аналіз існуючих підходів до використання технології комп'ютерного зору для розумного паркування.

Отже, після проведеного аналізу найефективнішими технологіями для розумного паркування на основі комп'ютерного зору виявилися дві технології на основі нейронних мереж - OpenCV та Google Cloud Vision API.

Одним з популярних засобів для розпізнавання зображень є бібліотека комп'ютерного зору OpenCV, яка є інфраструктурою для застосування методів CV в інформаційних системах. OpenCV використовується, серед іншого, для зміни розміру вхідних зображень, перетворення їх у векторну форму та виявлення особливостей цільових об'єктів у зображенні.

У той же час, одним з найпопулярніших підходів до виявлення особливостей зображення на сьогодні є глибоке навчання, зокрема CNN. Модель CNN поєднує багато функціональних операцій, які передають вхідне зображення як вектори ознак в результуючі дані, щоб оцінити належність ідентифікованих об'єктів до попередньо визначених класів.

Архітектура CNN, використана в цьому дослідженні, була взята з попередньої роботи авторів і зображена на рисунку 2.1. Відповідно до результатів класифікації в, можна зробити висновок, що поєднання системи GCV API та інструментів OpenCV + CNN може досягти більш надійної продуктивності та вищої точності класифікації.

Отже, для розробки кібер-фізичної системи розумного паркування було вирішено використовувати систему GCV API як детектор паркувальних місць та попередньо навчену ЗНМ як екстрактор функцій.

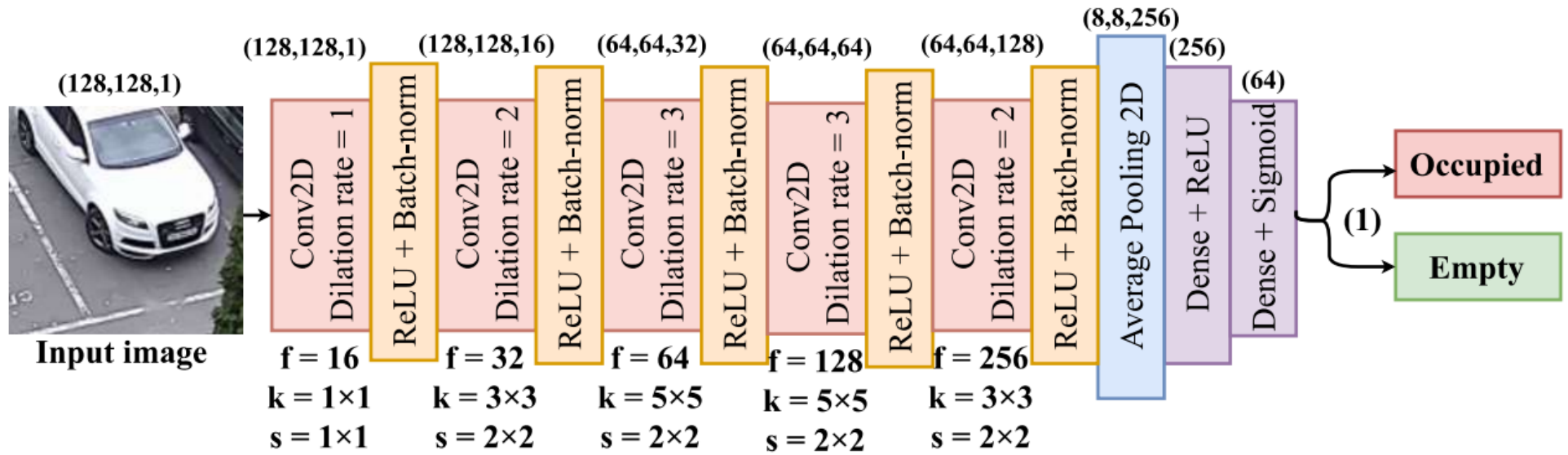


Рисунок 2.1. – Схема згорткової нейронної мережі, що використана в роботі

2.2. Вибір та налаштування алгоритму для навчання нейромережі.

Розглянемо систему для розумної парковки, яка складається з наступних компонентів апаратно-програмного забезпечення, яка зображена на рисунку 2.2

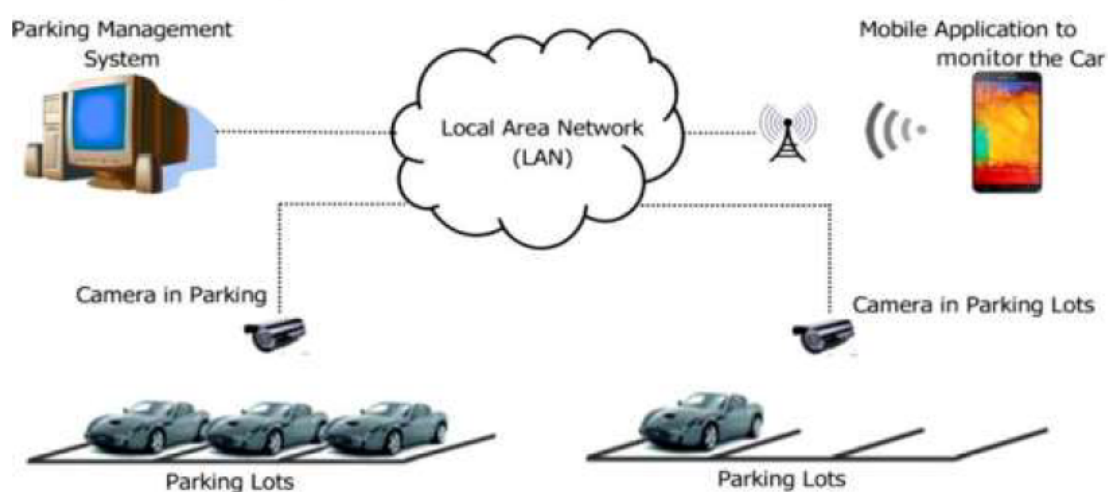


Рисунок 2.2. – Модель запропонованої системи Smart Parking.

З моделі видно, що камера є головним інструментом, який використовується для аналізу стану паркувального місця. Отже, виявлення вільних або зайнятих місць є ключовою задачею для розвитку системи Smart Parking. У сучасному світі глибоке навчання є важливою складовою розвитку рішень на основі бачення. Глибоке навчання - це підмножина машинного навчання, яка використовує нейронні мережі з кількома шарами для навчання на шаблонах. Незважаючи на останні досягнення глибокого навчання, є багато аспектів, які можна покращити в цій галузі. Глибоке навчання зазвичай використовує навчену модель для класифікації об'єктів. Нейронна мережа в глибокому навчанні складається з трьох основних шарів: вхідного, прихованого та вихідного. Вхідний шар приймає інформацію, у разі розпізнавання зображень - це будуть дані зображення. Потім вузли в

прихованому шарі зважуються, і наступний крок - обробка у вихідному шарі, що повертає результат класифікації на основі ваг. На Рисунку 2.3 представлено схематичне зображення нейронної мережі.

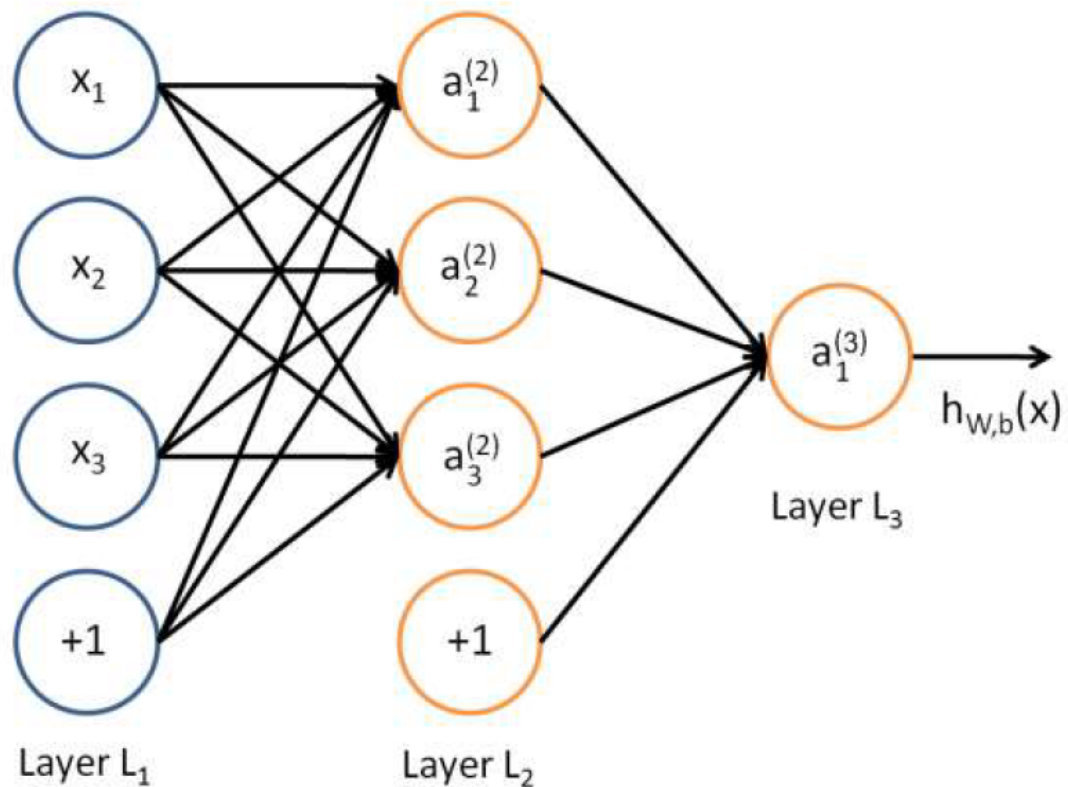


Рисунок 2.3. – Схема використовуваної нейронної мережі.

Під час розробки систем на основі обробки зображень, де необхідне розпізнавання об'єктів, згорткова нейронна мережа (CNN) може допомогти в ефективному виявленні шаблонів на зображеннях. Згідно з описом Amato та інших [5], використання великої кількості прихованих шарів у CNN дозволяє обробляти вхідні дані та надавати результат як вихід, покращуючи точність виявлення розпізнаних об'єктів.

CNN - це підклас нейронних мереж і є алгоритмом глибокого навчання, який зазвичай використовується для виявлення об'єктів на зображеннях. Це можливо завдяки тому, що архітектура CNN обробляє зображення як об'єми, що складаються з глибини, ширини та висоти. Передача даних між вхідним та прихованими шарами відбувається за допомогою нейронів, які

намагаються знайти закономірності в зображеннях шляхом фільтрації даних між шарами.

Для застосування CNN вимагається навчання даних, яке передбачає збір набору даних, що містить зображення потрібного об'єкта. Об'єкти на зображеннях визначаються та позначаються окремо. Після маркування набору даних він готовий для навчання. Під час навчання дані фільтруються в прихованих шарах, щоб знайти та вказати, які подібні шаблони поділяють усі зображення. Ці шаблони є параметрами, які забезпечують конкретний опис об'єкта/об'єктів та допомагають дізнатися, як виглядає конкретний об'єкт.

Тим не менш, навчання моделі для класифікації об'єктів може бути дуже дорогим з точки зору обчислень і часу. Це пов'язано з кількістю ітерацій, необхідних для того, щоб мережа знайшла і зрозуміла шари в нейронній мережі, що вимагає пристойного графічного блоку для виконання навчання.

Кластеризація є важливим методом у машинному навчанні, який використовується для класифікації об'єктів, які не мають позначок. Цей метод є методом навчання без нагляду, де дані не позначені, тому алгоритм кластеризації групує подібні дані в один кластер, використовуючи відстань між різними точками даних. Кластеризація є корисним інструментом для виявлення аномалій у даних та запобігання впливу викидів на кластери, які можуть виникати через вимірну помилку.

Обробка зображень передбачає інтерпретацію зображень, щоб отримати важливу інформацію, необхідну для досягнення певної мети. Цей процес можна порівняти зі сприйняттям людським органом зору, який допомагає людині отримати наочний образ світу та зрозуміти стан навколишнього середовища.

Метою обробки зображень є реалізація цієї ж функції в машинах. Сьогодні обробка зображень широко використовується в різних сферах, таких

як перевірка несправностей в електронних схемах, медична діагностика або відстеження об'єктів для цілей спостереження.

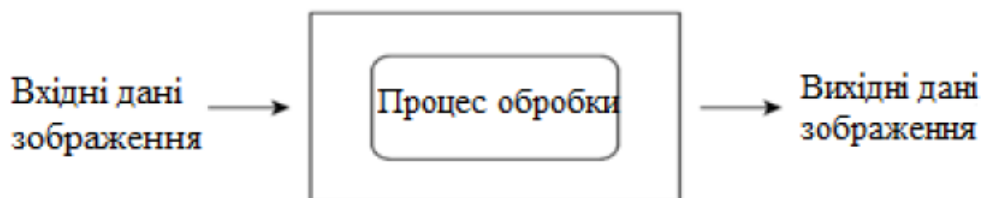


Рисунок 2.4. – Схема обробки зображення.

Пропонована система для розпізнавання стану паркувальних місць базується на технології глибокого навчання CNN. Ця техніка використовуватиметься для розробки програмного забезпечення, яке здатне розпізнавати зображення паркувальних місць та визначати їх стан (вільні чи зайняті).

Процес розпізнавання зображень в цій системі складається з таких етапів:

1. Зйомка зображення паркувального майданчика за допомогою камери;
2. Розділення зображення на сектори, кожен з яких відповідає одному паркувальному місцю;
3. Класифікація кожного сектора за допомогою навченої нейронної мережі, щоб визначити, чи є паркувальне місце вільним чи зайнятим.

Хоча в галузі глибокого навчання вже були досягнуті значні успіхи, є ще багато можливостей для його вдосконалення. Останнім часом техніка глибокого навчання широко використовується для розпізнавання об'єктів на основі бачення. Принцип роботи глибокого навчання полягає у використанні навченої моделі для класифікації об'єктів.

Таким чином, в результаті аналізу було визначено, що найбільш ефективними для цієї системи є дві нейромережі - OpenCV та Google Cloud Vision API.

2.3. Вибір моделі та аналіз вимог до системи парковки на основі штучної нейронної мережі.

Системи паркування з часом стають все більш популярними, оскільки вони дозволяють ефективніше використовувати обмежені просторові ресурси та зменшувати час на пошук вільних місць для паркування. Однак, розробка ефективної системи паркування може бути складною задачею, оскільки вона повинна враховувати різноманітні фактори, такі як розмір транспортних засобів, типи парковок, розміщення камер та сенсорів, і багато іншого.

У цьому контексті штучні нейронні мережі можуть бути корисними для розробки системи паркування, оскільки вони здатні аналізувати складні дані та навчатися з досвіду. Проте, вибір відповідної моделі штучної нейронної мережі є ключовим фактором у розробці ефективної системи паркування.

У даній роботі буде розглянуто Google Cloud Vision для визначення вимог до системи паркування на основі штучної нейронної мережі.

Google Cloud Vision є однією з послуг, наданих Google Cloud Platform, яка надає можливості комп'ютерного зору та обробки зображень за допомогою штучного інтелекту. Вона дозволяє розпізнавати, аналізувати та виконувати різноманітні завдання обробки зображень.

Основні можливості Google Cloud Vision включають:

- **Розпізнавання об'єктів:** Система може ідентифікувати об'єкти на зображеннях, включаючи розпізнавання лиць, міток, логотипів, автомобільних номерних знаків та іншого.
- **Класифікація зображень:** Можливість класифікувати зображення на основі зарані визначених категорій, таких як животні, пейзажі, об'єкти тощо.
- **Виявлення облич:** Можливість виявлення облич на зображеннях, включаючи розпізнавання різних різнобічних характеристик, наприклад, емоцій, виразів обличчя, розташування ключових точок.

- Оптичне розпізнавання символів (OCR): Можливість витягувати текстову інформацію зі зображень, включаючи розпізнавання тексту з фотографій, сканованих документів, банківських карток тощо
- Аналіз зображень: Можливість аналізувати зображення з точки зору безпеки, визначення вікової категорії, розпізнавання сцен, оцінки належності до певних категорій тощо.
- Виявлення розміщення зображення: Можливість визначення географічного розміщення зображення на основі візуальних ознак.
- Обробка зображень: Google Cloud Vision надає можливість виконувати різноманітні операції обробки зображень. Ви можете змінювати розмір зображення, обрізати його, вирівнювати, застосовувати ефекти, змінювати контрастність та яскравість, видаляти шуми тощо. Це дозволяє покращити якість зображення та підготувати його для подальшого використання.

Google Cloud Vision може бути використана в багатьох сферах, включаючи медіа, рекламу, безпеку, електронну комерцію, медицину та інші. Вона надає розширені можливості обробки зображень, що дозволяє виконувати складні завдання комп'ютерного зору за допомогою штучного інтелекту.

Процес розробки додатку з використанням Google Cloud Vision може включати наступні кроки:

1. Реєстрація та налаштування проекту: Спочатку вам потрібно зареєструватися на Google Cloud Platform і створити новий проект. Потім ви повинні ввімкнути службу Google Cloud Vision для свого проекту.

2. Отримання API-ключа: Для взаємодії з Google Cloud Vision API вам потрібен API-ключ. Ви можете отримати його в панелі керування Google Cloud Platform.

3. Встановлення SDK або бібліотеки: Виберіть мову програмування, з якої ви плануєте розробляти додаток. Google Cloud Vision надає SDK та

бібліотеки для різних мов програмування, таких як Python, Java, Node.js та інші. Встановіть необхідні інструменти та залежності для розробки.

4. Налаштування автентифікації: Ви повинні налаштувати автентифікацію для взаємодії з Google Cloud Vision API. Використовуйте свій API-ключ або інші методи аутентифікації, такі як службові облікові записи, для доступу до API.

5. Взаємодія з Google Cloud Vision API: Використовуйте функції та можливості Google Cloud Vision API у вашому додатку. Наприклад, ви можете надсилати зображення для розпізнавання об'єктів, класифікації, аналізу змісту або виявлення облич.

6. Обробка результатів: Після отримання результатів від Google Cloud Vision API ви можете обробити їх у вашому додатку. Це може включати аналіз отриманих міток, обробку розпізнаного тексту або відображення виявлених облич на зображеннях.

7. Тестування та налагодження: Перевірте роботу вашого додатку, виконавши тестування та налагодження. Переконайтеся, що взаємодія з Google Cloud Vision API працює належним чином і повертає очікувані результати.

Під час проектування системи для розумної парковки на основі штучної нейронної мережі були сформульовані основні нефункційні вимоги для цієї системи. Відповідна таблиця з цими вимогами наведена у таблиці 2.2.

Назва вимоги	Опис вимоги
Надійність	Система повинна забезпечувати надійну роботу, без аварій, із забезпеченням резервування і відновлення роботи в разі відмови.
Безпека	Система повинна бути безпечною для користувачів, уникати можливих загроз життю і здоров'ю людей та ушкодження майна.
Ефективність	Система повинна бути ефективною в використанні ресурсів, забезпечуючи оптимальну роботу та швидкий

	відгук на запити користувачів.
Масштабованість	Система повинна бути готовою до масштабування, щоб вона могла розширюватися та збільшувати свою потужність для задоволення зростаючих потреб користувачів.
Простота використання	Система повинна бути легкою у використанні та зрозумілою для користувачів різного рівня технічної підготовки.
Сумісність	Система повинна бути сумісною з іншими системами та пристроями, що можуть бути використані для взаємодії з нею.
Адаптованість	Система повинна бути готовою до адаптації до змінних умов та потреб користувачів.
Споживання енергії	Система повинна бути енергоефективною, щоб зменшити витрати на електроенергію та мінімізувати вплив на навколишнє середовище.

Таблиця 2.2. - Нефункційні вимоги до системи для проекрованої системи розумної парковки на основі штучної нейронної мережі

Під час проектування програмного забезпечення для розумної парковки були сформульовані основні функційні вимоги, які включають:

- Функціонал огляду панорами парковки.
- Список доступних парковок для перегляду.
- Можливість спостереження за автомобілем в режимі реального часу.
- Відображення вільних та зайнятих паркомісць у зручному для користувача вигляді.

РОЗДІЛ 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

3.1. Архітектура системи для прокладання візуалізації та візуалізації маршрутів на основі технології доповненої реальності.

Для розробки прототипу системи для розумного паркування необхідно спочатку дослідити необхідні функції та вимоги, які є важливими для цієї системи. Це дослідження дозволить зрозуміти процес розробки всієї системи. Для встановлення камери в загальнодоступній зоні потрібно отримати дозвіл на камеру відеоспостереження. Важливим фактором є проблема конфіденційності, яка виникає при використанні рішень на основі бачення. Надсилання зображень на сервер або хмару через Інтернет також супроводжується проблемами безпеки. Однак, можна використовувати певний процес шифрування для вирішення проблем безпеки. Виконання завдань з обробки зображень у місці, де зображення фіксуються без збереження або надсилання будь-яких даних зображення через Інтернет, може допомогти уникнути цих проблем. Тому, аспекти конфіденційності та безпеки слід враховувати як вимоги для інтелектуального рішення для паркування. На основі аналізу вимог до системи для розумного паркування з використанням розпізнавання зображень, було розроблено діаграму вимог, яка представлена на рисунку 3.1.

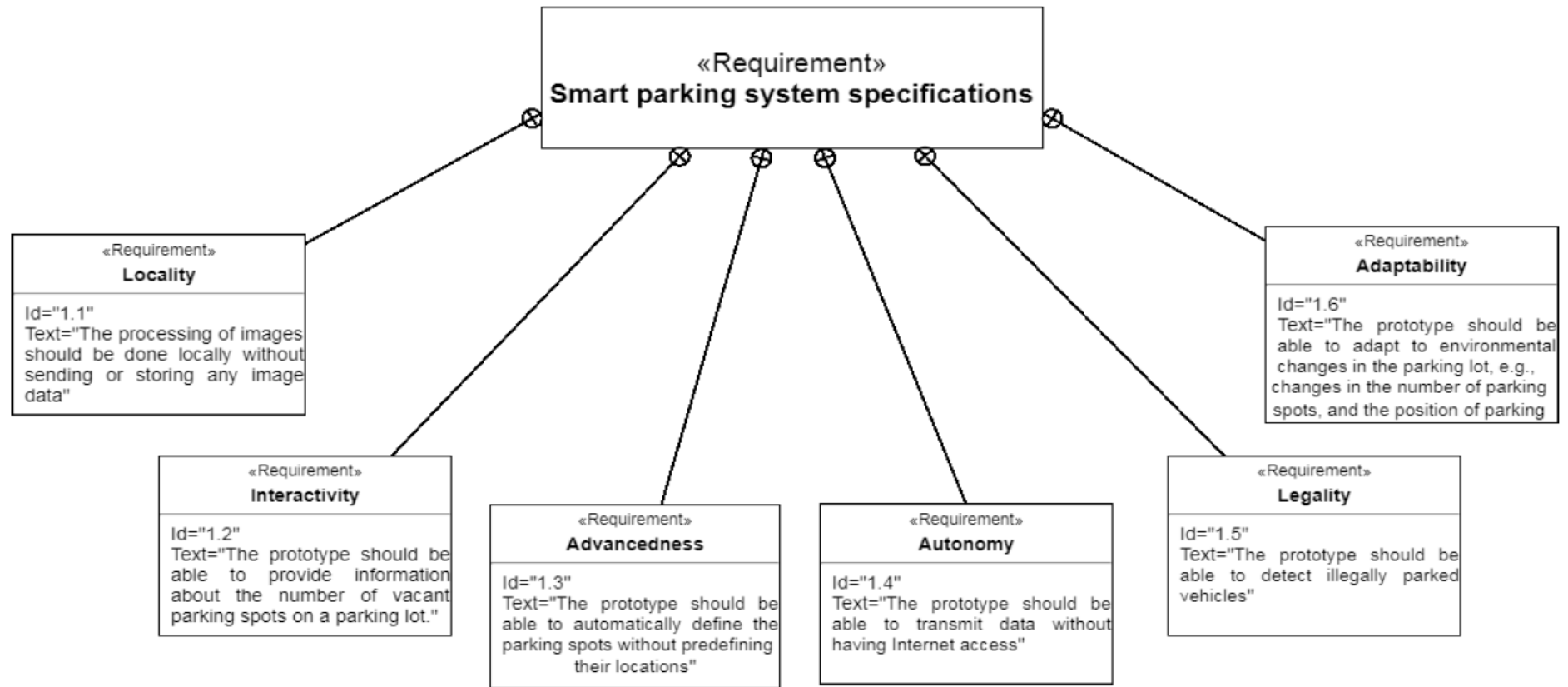


Рисунок 3.1. – Діаграма вимог до системи розумної парковки.

Створення розумної системи паркування на основі комп'ютерного бачення включає різні підсистеми, такі як хмара, платформа візуалізації та база даних. Головною метою дослідження є створення прототипу, який забезпечує захоплення зображень, вилучення інформації про зайнятість паркувальних місць з цих зображень та надсилання цієї інформації на платформу невеликого розміру. Вибір найкращої технології передачі даних, платформи візуалізації та зберігання даних виходить за рамки даного дослідження.

Для отримання загального уявлення про різні підпроблеми системи було розроблено потік розбивки проблеми на основі діаграми вимог, яка була представлена вище. Кожна підзадача містить набір завдань, необхідних для вирішення підзадачі. З метою вирішення основної проблеми були визначені чотири підзадачі: виявлення місця для паркування, передача даних, вибір і конфігурація обладнання та інтеграція.

Головним напрямком дослідження є вирішення підпроблеми виявлення місця паркування. Дерево декомпозиції задачі побудови кіберфізичної системи для розумної парковки зображене на рисунку 3.2.

Згідно з рисунком 3.2, основні етапи дослідження полягають у виконанні підзадач, виділених червоними стрілками.

Однією з таких підзадач є виявлення місця для паркування з метою побудови динамічної розумної системи паркування. Для цього місця для паркування мають бути визначені автоматично. Це завдання можна виконати за допомогою алгоритму виявлення об'єктів. Вибраний підхід для виявлення паркувальних місць заснований на технології глибокого навчання, яка дозволяє виявляти транспортні засоби. Цей підхід обраний через його простоту виявлення паркувальних місць на стоянці.

Є ще один підхід, який можна використовувати для виявлення місць для паркування, - це визначення лінії. Однак, у цього підходу є деякі недоліки, такі як можливість стирання або розмивання ліній або їх покривання снігом.

Метою передачі даних в системі є передача даних на сервер, де можна зареєструвати інформацію про паркування. Передані дані мають на меті надати інформацію про стан паркінгу, тобто про те, скільки паркувальних місць доступно і скільки з цих місць для паркування не зайнято.

Для роботи системи потрібне обладнання, яке складається з трьох частин: камери, комп'ютера та інтерфейсу користувача. Камера призначена для зйомки зображень стоянки та відправки їх на бортовий комп'ютер. Комп'ютер в свою чергу виконує обчислення та обробку зображень, отриманих від камери, для виявлення місць для паркування.

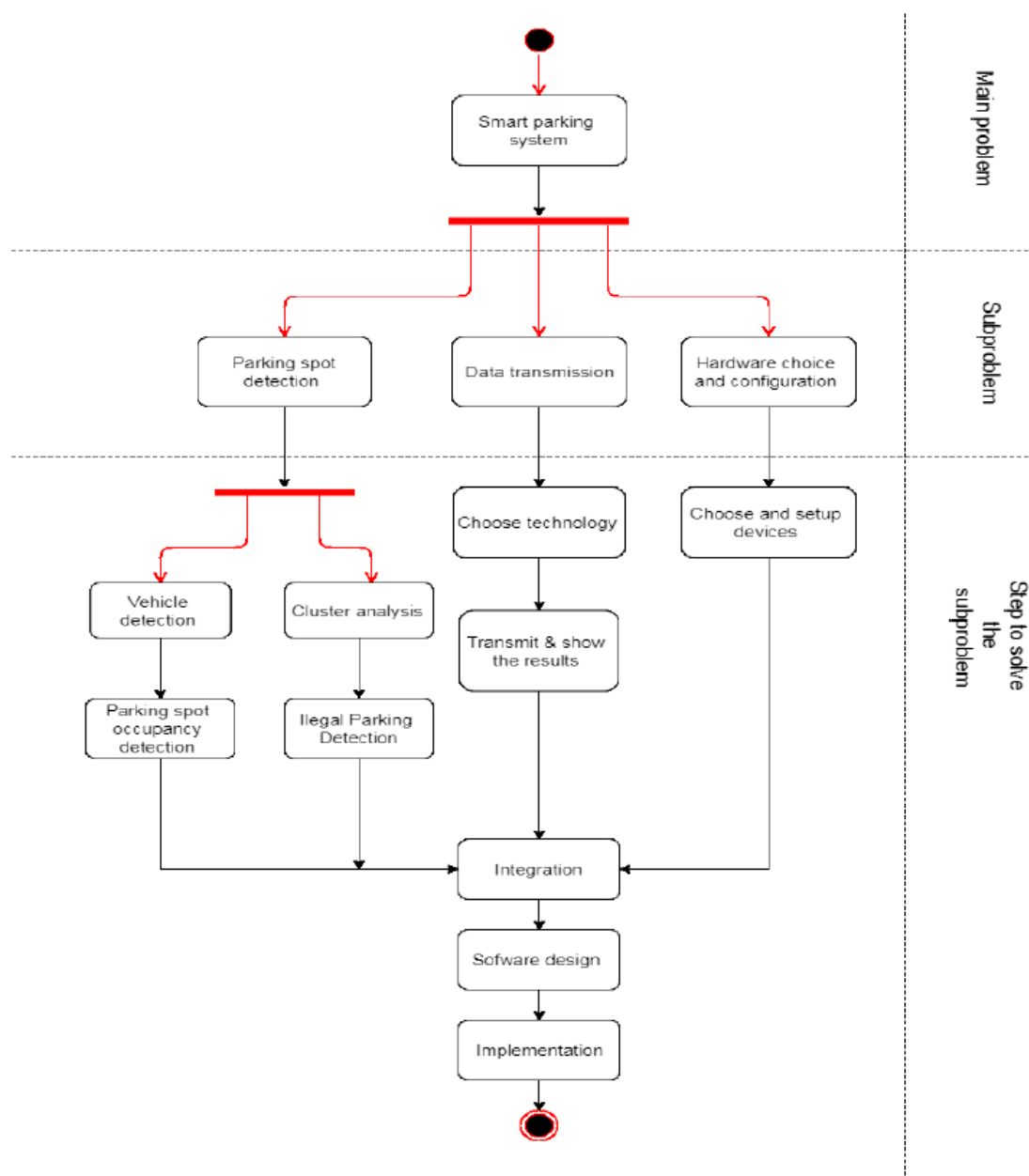


Рисунок 3.2. – Дерево декомпозиції задачі побудови системи для парковки.

Для передачі інформації про автостоянку на сервер, таку як ідентифікатор парковки, кількість доступних місць та кількість незайнятих, використовується пристрій передачі даних. Частина інтерфейсу користувача можна розробити у вигляді мобільного додатка. Щоб інтегрувати різні частини (виявлення місця для паркування, реалізацію обраної техніки передачі даних та вибір апаратного забезпечення) в одну систему, потрібно провести інтеграцію.

Коли користувач відкриває мобільний додаток для розумної парковки, перед ним відкривається екран з картою міста. На цьому екрані можна вибрати парковку, яку користувач хоче перевірити на наявність вільних паркомісць. Після вибору парковки користувач переходить на екран, на якому показані вільні (зеленим кольором) та зайняті (показані у вигляді іконки автомобіля) місця для паркування у вигляді схематичної розмітки. Після вибору потрібного місця користувач направляється на парковку. Крім схематичного зображення вільних та зайнятих паркомісць, користувач може подивитись відео з камери спостереження у режимі реального часу з метою безпеки автомобіля.

Для розумної парковки використовується кіберфізична система, що складається з серверної та клієнтської частин. Серверна частина отримує зображення з камер зовнішнього спостереження та обробляє його з метою ідентифікації автомобілів на парковках. Після обробки зображення, сервер передає клієнтській частині схематичне зображення з вільними та зайнятими паркомісцями, яке буде доступне для користувача в мобільному додатку. Важливо регулярно оновлювати дані та передавати їх у клієнтську частину, щоб забезпечити правильну роботу системи. Схема роботи кіберфізичної системи для розумної парковки, що базується на технології розпізнавання

зображень з використанням штучної нейронної мережі, зображена на рисунку 3.3.

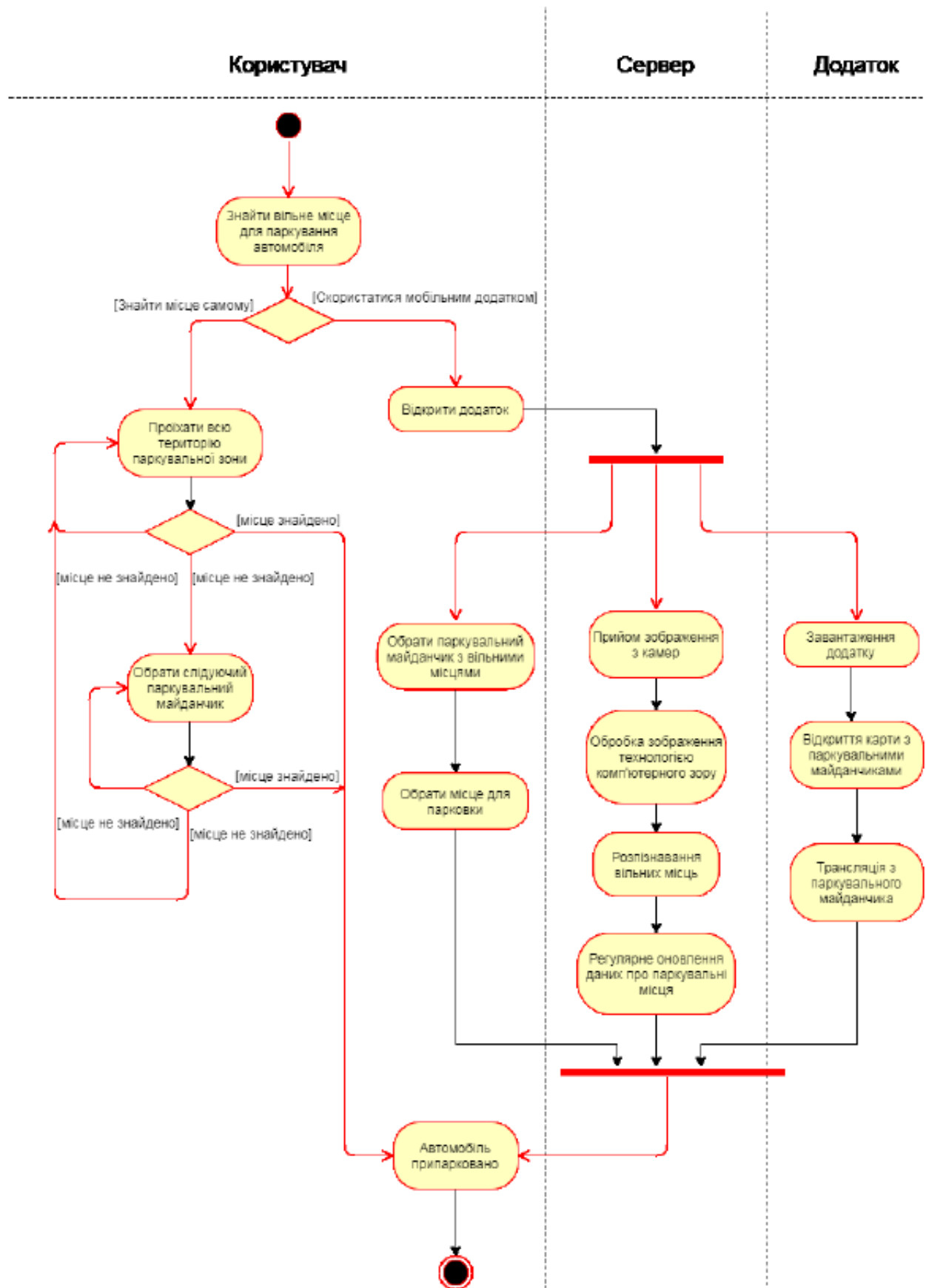


Рисунок 3.3. – Схема роботи системи для розумної парковки на основі технології розпізнавання зображень

На рисунку 3.4 представлено схему взаємодії серверної та клієнтської частин кіберфізичної системи для розумної парковки на основі технології розпізнавання зображень з використанням штучної нейронної мережі. У верхньому лівому кутку діаграми зображений сервер обробки зображень Google Cloud Vision API, який був вибраний після експериментів у розділі 2. У нижньому лівому кутку зображений клієнтський пристрій - смартфон з операційною системою IOS не нижче версії 14 з встановлений додатком ParkVision. У центрі зображений фізичний сервер на базі операційної системи Linux та веб-сервер на AWS EC2.

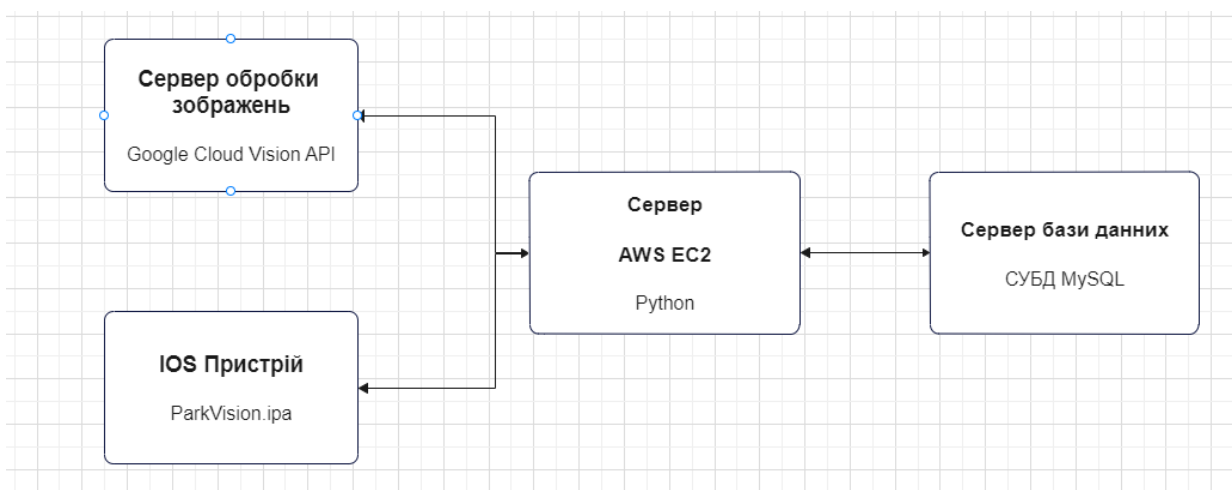


Рисунок 3.4. – Схема взаємодії апаратного та програмного забезпечення кіберфізичної системи для розумної парковки з використанням штучної нейронної мережі.

3.2. Опис розробленої системи знаходження вільних місць парковки.

Для демонстрації роботи кіберфізичної системи для розумної парковки на основі розпізнавання зображень автомобілів з використанням штучної нейронної мережі, буде використана ширококутна камера зовнішнього спостереження, кут огляду якої охоплюватиме весь парковий майданчик. Смуга розмітки паркового майданчика має бути чітко видимою та

накресленою з дотриманням стандартів. У даній роботі запропонована схема паркового майданчика, яка представлена на рисунку 3.5.



Рисунок 3.5. – Запропонована у роботі схема паркомайданчика.

Для оснащення паркомайданчика потрібне наступне апаратне забезпечення:

- PoE-коммутатор TP-LINK TL-SG1005LP.
- IP відеокамера Dahua DH-IPC-HDW2431TP-AS-S2.
- Витя пара зовнішньої прокладки UTP c.5E 4 x 2 x 0.51 мм² 25 м.
- Кабель КППЕТ-ВП (100) 4*2*0,51 (FTP-cat.5E), ОК-net, CU, ізоляція ПЕ, екр., з трос. 7 * 0,5 для нар. роб.,305м.
- 2x Спліттер PoE для CCTV камер Ethernet 5.5x2.1мм.
- Конектор Cor-X RJ45 connector CAT5E 8P8C 10 шт.
- Інжектор PoE-INJECTOR Lite.

В таблиці 3.1 наведено перелік апаратного забезпечення, яке використовувалось для оснащення розумного паркомайданчика та його функційного призначення.

Компонент	Функціонал
Камера зовнішнього спостереження	Збір зображень паркомайданчика, що передаються на сервер для обробки
Сервер обробки зображень	Обробка зображень з камери зовнішнього спостереження з метою ідентифікації автомобілів, передача інформації на

	клієнтську частину
Фізичний сервер	Забезпечення функціонування веб-сервера та сервера баз даних
Веб-сервер	Надання доступу до системи для користувачів, обробка запитів
Сервер баз даних	Зберігання та оновлення даних про паркомайданчик, автомобілі та їх власників
Клієнтський пристрій	Пристрій користувача, який відображає інформацію про стан паркомайданчика та доступ до функцій системи, таких як бронювання місця для паркування або сплачування плати за користування паркомайданчиком. Може бути різних типів: смартфон, планшет, комп'ютер тощо.

Таблиця 3.1 Опис функціоналу апаратного забезпечення для оснащення розумного паркомайданчика.

Для розробки серверної частини кіберфізичної системи для розумної парковки на основі технології розпізнавання зображень з використанням штучної нейронної мережі було використано такі програмні засоби:

- 1) сервер обробки зображень Google Cloud Vision API;
- 2) клієнтський пристрій - будь-який смартфон з операційною системою не старішою, ніж iOS 14;
- 3) фізичний сервер на базі операційної системи Linux;
- 4) веб-сервер на AWS EC2;
- 5) сервер бази даних з СУБД MySQL.

Для розробки клієнтської частини системи для розумної парковки на основі технології розпізнавання зображень з використанням штучної нейронної мережі у вигляді мобільного додатку ParkVision буде використано такі програмні засоби як операційну систему iOS, мову програмування Python та середовище для програмування IDLE.

Опис основних функцій програмного забезпечення подано в зведеній таблиці 3.2.

Основна функція	Опис
<code>client = vision_v1.ImageAnnotatorClient()</code>	Створення клієнта Google Cloud Vision API для розпізнавання зображень
<code>image = types.Image(content=content)</code>	Створення об'єкту зображення з завантаженого файлу
<code>response = client.annotate_image({...})</code>	Виконання анотування зображення та отримання результатів розпізнавання об'єктів
<code>localized_object_annotations = response.localized_object_annotations</code>	Збереження списку анованих об'єктів
<code>with Image.open(file_path) as img: draw = ImageDraw.Draw(img) ...</code>	Відкриття зображення та створення об'єкту для малювання
<code>for obj in localized_object_annotations: box = [(vertex.x, vertex.y) for vertex in obj.bounding_poly.normalized_vertices] draw.rectangle(box, outline='green', width=3)</code>	Проходження по списку анованих об'єктів та малювання прямокутних рамок для кожного об'єкту
<code>img.save(f' {file_path}-annotated.jpg')</code>	Збереження нового зображення з нарисованими рамками в файл.

Таблиця 3.2 Опис основних функцій ПЗ

Також у ході виконання роботи було розроблено дизайн прототипу мобільного додатку для розумної парковки на основі технології розпізнавання зображень для операційної системи iOS(рис. 3.6 – 3.8).



Рисунок 3.6. – Вікно завантаження додатку.

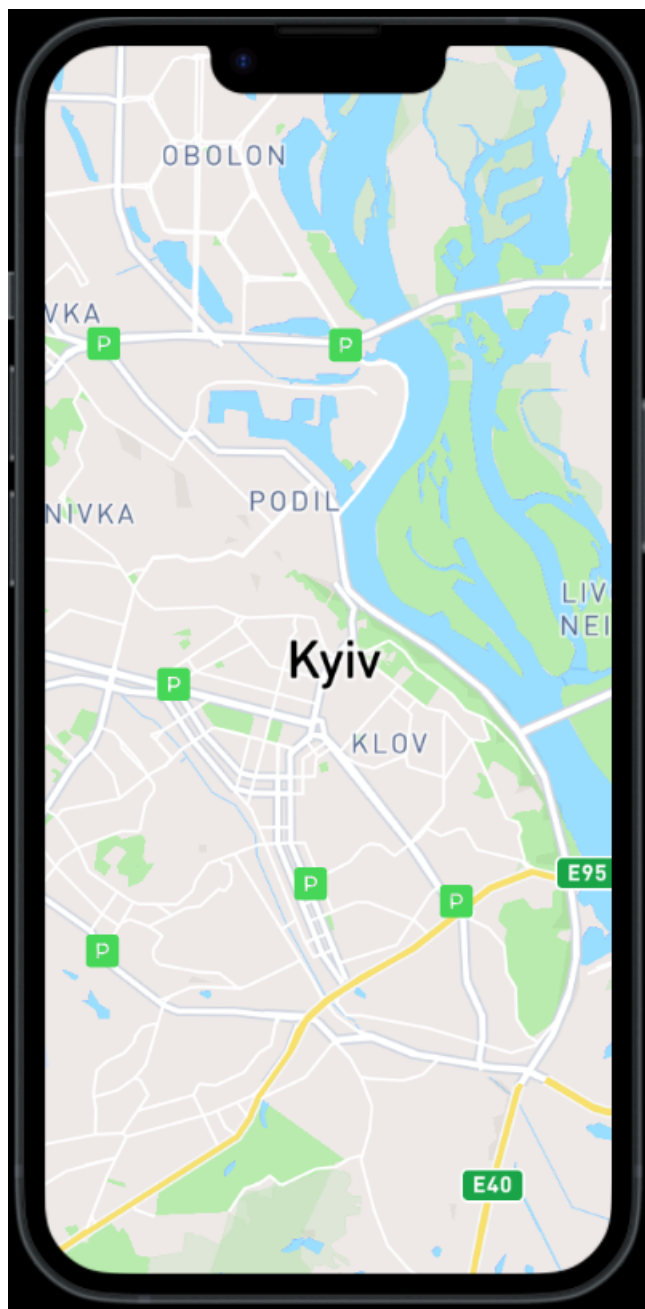


Рисунок 3.7. – Карта додатку для місце знаходження парковки.



Рисунок 3.8. – Результат роботи додатку(знаходження вільного місця).

3.3. Порівняння результатів роботи розробленої системи з результатами інших досліджень.

Для порівняння створених додатків і розробкою дипломного проекту використаємо таблицю(таб 3.2.).

Назва додатку	Функції	Операційна система	Вартість
ParkPow	Знаходить вільні місця на парковці та надає користувачеві	Android, iOS	Безкоштовний

	відповідну інформацію.		
ParkMobile	Знаходить вільні місця на парковці, дозволяє користувачам забронювати парковку та оплачувати її.	Android, iOS	Безкоштовний, або \$1.50 за транзакцію
EasyPark	Знаходить вільні місця на парковці, дозволяє користувачам забронювати парковку та оплачувати її.	Android, iOS	Безкоштовний, або \$0.25 за транзакцію
ParkWhiz	Знаходить вільні місця на парковці, дозволяє користувачам забронювати парковку та оплачувати її.	Android, iOS	Безкоштовний, або від \$1.99 за транзакцію
ParkVision	Знаходить вільні місця на парковці підключаючись до камер відеоспостереження та обробляє зображення за допомогою нейромережі.	iOS	Безкоштовний

Таблиця 3.2. - порівнянням додатків для знаходження вільних місць на парковці

Ці додатки мають схожі функції, але можуть відрізнитися у вартості та в підтримуваній операційній системі. Якщо вартість є важливим фактором для користувача, він може обрати додаток, який коштує менше або пропонує оплату за транзакцію. Якщо користувач має певну операційну систему, він повинен знайти додаток, який підтримує його ОС.

ВИСНОВКИ

В даній роботі було проведено теоретичні та практичні дослідження з метою удосконалення методів розпізнавання автомобілів та розробки інформаційної моделі для системи для розумної парковки на основі технології розпізнавання зображень з використанням штучної нейронної мережі.

У першому розділі було проведено аналіз галузі розумної парковки. Також проведено огляд літературних джерел та порівняльний аналіз існуючих рішень із застосуванням методів розпізнавання зображень. Зроблено висновок, що задача розпізнавання зображень автомобілів, отриманих з камер зовнішнього спостереження, наразі є актуальною.

У другому розділі було проведено аналіз інформаційних технологій розпізнавання зображень на основі комп'ютерного зору. Була обрана технологія Google Cloud Vision для розробки системи розумного паркування на основі технології комп'ютерного зору. Розроблено нову модель на основі тонко налаштованої згорткової нейронної мережі для виявлення порожніх і зайнятих слотів на зображеннях парковок, зібраних з набору даних. Досягнуті результати свідчать про підвищення ефективності виявлення парковок та покращення їх точності. Зроблено висновок про те, що технологія Google Cloud Vision як детектор паркувальних місць і попередньо навчена згортка нейронна мережа як екстрактор і класифікація функцій були прийняті для розробки системи паркування.

У третьому розділі описано інформаційну модель, що базується на згортковій нейронній мережі для розпізнавання зображень, використану в системі для розумної парковки.

Подальше дослідження було спрямоване на розробку серверної та клієнтської частин у вигляді мобільного додатку, який буде відслідковувати наявність вільних місць на парковці. Отримані результати включають розробку апаратно-програмної інформаційної системи для розумної парковки

на основі технології розпізнавання зображень та її реалізацію у вигляді серверної частини програмного забезпечення для мобільного додатка на базі ОС iOS.

Розроблений додаток для розумної парковки має декілька переваг порівняно з іншими додатками:

- використання Google Cloud Vision дозволяє забезпечити високу точність розпізнавання транспортних засобів, що дозволяє заощадити час та зробити процес паркування більш зручним
- має інтуїтивно зрозумілий і простий інтерфейс, що робить його використання простим та зручним для користувачів.
- використання Google Cloud Vision забезпечує збереження та обробку даних у безпечному та захищеному середовищі, що є дуже важливим для користувачів та забезпечує захист їх приватності.

Отже, використання Google Cloud Vision та додаткова функціональність додатку розумної парковки роблять його більш зручним та ефективним порівняно з іншими додатками на ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Automotive market research institute. Analytical study of the secondary car market of Ukraine. Eauto.org.ua. URL: <https://eauto.org.ua/news/13-analitichne-doslidzhennya-vtorinnogo-avtorink-u-ukrajini> (дата звернення: 30.12.2021).
2. Pavlova O., Kovalenko V., Novorushchenko T., Avsiyevych V. Neural network based image recognition method for smart parking. *Comput. Syst. Inf. Technol.* 3, 2021. pp. 49–55
3. Авсієвич В., Коваленко В. Аналіз інформаційних технологій для розумної парковки на основі штучних нейронних мереж. *Актуальні Проблеми Комп'ютерних Наук (АПКН-2021)*, Хмельницький, Україна, 15-16 жовтня 2021. Хмельницький: ХНУ, 2021. С. 12-14.
4. Barmak O., Radiuk P. Web-based information technology for classifying and interpreting early pneumonia based on fine-tuned convolutional neural network. *Comput. Syst. Inf. Technol.* 3, 2021. pp. 12–18 DOI:10.31891/CSIT-2021-3-2
5. Google Cloud Vision API. Майбутнє Computer Vision as a service настало? URL: <https://habr.com/post/312714/> (дата звернення: 13.03.2022).
6. Build your own Vehicle Detection Model using OpenCV and Python. URL: <https://www.analyticsvidhya.com/blog/2020/04/vehicle-detection-opencv-python/>(дата звернення: 13.03.2022).
7. Khan S.D., Ullah H. A survey of advances in vision-based vehicle re-identification. *Comput. Vis. Image Underst.* 182, 2019. pp. 50–63
8. Dixit M., Srimathi C., Doss R., Loke S., Saleemdurai M.A. Smart parking with computer vision and IoT technology. In: 2020. 43rd International Conference on Telecommunications and Signal Processing (TSP-2020). IEEE Inc., Milan, Italy, 7-9 July 2020. pp. 170–174.

9. Do H., Choi J.Y. Context-based parking slot detection with a realistic dataset. *IEEE Access*. 8, 171551–171559. 2020.
10. Li W., Cao H., Liao J., Xia J., Cao L., Knoll A. Parking slot detection on around-view images using DCNN. *Front. Neurorobot*. 14, 2020.
11. Trivedi J., Devi M.S., Dhara D. Canny edge detection based real-time intelligent parking management system. *Sci. J. Silesian Univ. Technol. Transp.* 106, 2020. pp.197–208.
12. Noor M., Shrivastava A. Automatic parking slot occupancy detection using Laplacian operator and morphological kernel dilation. In: 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT-2021). pp. 825–831. IEEE Inc., Bhopal, India, 18-19 June 2021 DOI: 10.1109/CSNT51715.2021.9509620
13. Hakim I.M., Christover D., Jaya Marindra A.M. Implementation of an image processing based smart parking system using Haar-Cascade method. In: 2019 IEEE 9th Symposium on Computer Applications Industrial Electronics (ISCAIE-2019). pp. 222–227. IEEE Inc., Penang, Malaysia, 27-28 April 2019. DOI:10.1109/ISCAIE.2019.8743906
14. Tang J.S.L., Manickam S. Parking lot occupancy detection using image overlay and intersection technique with Harris corner detector. *J. Eng. Technol.* 11, 2020. pp.37–52
15. Sun R., Wang G., Zhang W., Hsu L.-T., Ochieng W.Y. A gradient boosting decision tree based GPS signal reception classification algorithm. *Appl. Soft Comput.* 86, 105942. 2020.
16. Zhang L., Huang J., Li X., Xiong L. Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset. *IEEE Trans. Image Process.* 27, 5350–5364. 2018.
17. Li W., Cao L., Yan L., Li C., Feng X., Zhao P. Vacant parking slot detection in the around view image based on deep learning. *Sensors*. 20, 2138. 2020.

18. Trivedi J.D., Sarada Devi M., Dave D.H. Different modules for car parking system demonstrated using Hough transform for smart city development. *Intelligent Manufacturing and Energy Sustainability*. Springer Singapore, Singapore, 2020. pp. 109–121.
19. Manjula G., Govinda Rajulu G., Anand R., Thirukrishna J.T. Implementation of smart parking application using IoT and machine learning algorithms. In: Smys, S., Bestak, R., Palanisamy, R., and Kotuliak, I. (eds.) *Computer Networks and Inventive Communication Technologies*. Springer Singapore, Singapore, 2022. pp. 247–257.
20. Min C., Xu J., Xiao L., Zhao D., Nie Y., Dai, B. Attentional graph neural network for parking-slot detection. *IEEE Robot. Autom. Lett.* 6, 3445–3450, 2021.
21. Gollapudi S. *OpenCV with Python*. In: *Learn Computer Vision Using OpenCV*. Apress, Berkeley, CA, 2019. pp. 31–50.
22. Krizhevsky A., Sutskever I., Hinton E.G. ImageNet classification with deep convolutional neural networks. *Commun. ACM.* 60, 2017. pp. 84–90.
23. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. In: *3rd International Conference on Learning Representations (ICLR-2015)*. pp. 1–14. International Conference on Learning Representations, ICLR, San Diego, CA, USA, 7-9 May 2015.
24. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L. MobileNetV2: Inverted residuals and linear bottlenecks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-2018)*. pp. 4510–4520. IEEE Inc., Salt Lake City, UT, USA, 18–23 June 2018.
25. De Almeida, P.R.L., Oliveira, L.S., Britto A.S., Silva E.J., Koerich A.L.: PKLot - A robust dataset for parking lot classification. *Expert Syst. Appl.* 42, 4937–4949, 2015.

26. Cesana A.E. Redondi M. Tagliasacchi L. Baroffio, L. Bondi. A visual sensor network for parking lot occupancy detection in smart cities. IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015.
27. Vandelinde H., Milligan S., Cavanagh M., Understanding Ultrasonic Level Measurement. SIEMENS, 2013.
28. Zhi-yuan Z., He R., Jie T. A method for optimizing the position of passive UHF RFID tags. 2018 IEEE International Conference on RFID-Technology and Applications, 2018.
29. Amato G., Carrara F., Falchi F., Gennaro C., Meghini C., and Vairo C. Deep learning for decentralized parking lot occupancy detection. Expert Systems with Applications, 72:327 –334, 2017.
30. Hurwitz J. and Kirsch D. Machine Learning For Dummies - IBM Limited Edition. John Wiley & Sons, Inc., New Jersey, NJ, USA, 2018. ISBN 9781119454946.
31. Shanmugamani R. Deep Learning for Computer Vision. Packt Publishing Limited, Birmingham, UK, 2017. ISBN 9781788295628.

**ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
КІБЕРФІЗИЧНОЇ СИСТЕМИ ДЛЯ РОЗУМНОЇ ПАРКОВКИ НА
ОСНОВІ ТЕХНОЛОГІЇ КОМП'ЮТЕРНОГО ЗОРУ**

```
import io
from PIL import Image, ImageDraw
from google.cloud import vision_v1
from google.cloud.vision_v1 import types
def detect_objects(file_path):
    # Create client and load image into memory
    client = vision_v1.ImageAnnotatorClient()
    with io.open(file_path, 'rb') as image_file:
        content = image_file.read()
    image = types.Image(content=content)
    # Perform object localization annotation
    response = client.annotate_image({
        'image': image,
        'features': [{ 'type':
vision_v1.enums.Feature.Type.OBJECT_LOCALIZATION} ]
    })
    localized_object_annotations = response.localized_object_annotations
    # Draw bounding boxes on image for each detected object
    with Image.open(file_path) as img:
        draw = ImageDraw.Draw(img)
        for obj in localized_object_annotations:
            box = [(vertex.x, vertex.y) for vertex in
obj.bounding_poly.normalized_vertices]
            draw.rectangle(box, outline='green', width=3)
    img.save(f' {file_path}-annotated.jpg')
```