

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**

В.о. завідувача кафедри інтелектуальних  
інформаційних систем, канд. техн. наук, доц.

\_\_\_\_\_ Є. В. Сіденко

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

**ІНТЕЛЕКТУАЛЬНА СИСТЕМА РОЗПІЗНАВАННЯ ТА**  
**ПОДОЛАННЯ ПЕРЕШКОД ТРАНСПОРТНИМИ**  
**ЗАСОБАМИ**

Спеціальність 122 «Комп'ютерні науки»

**122 – МКР – 601.21920103**

*Виконав студент 6-го курсу, групи 601*

\_\_\_\_\_ **О. М. Миронюк**

*(підпис, ініціали та прізвище)*

«15» лютого 2023 р.

*Керівник: канд. техн. наук, доцент*

\_\_\_\_\_ **Є. В. Сіденко**

*(підпис, ініціали та прізвище)*

«15» лютого 2023 р.

**Миколаїв – 2023**



5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: опис основних питань охорони праці пов'язаних з професійною діяльністю та убезпечення працівників при надзвичайних ситуаціях

7. Консультанти:

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Григор'єва Л.І. докт. біол. наук, професор	
Методична частина	Сіденко Є. В. канд. техн. наук, доц.	

Керівник роботи канд. техн. наук, доц. Сіденко Є. В.  
(наук. ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Завдання прийнято до виконання Миронюк О. М.  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

Дата видачі завдання «07» листопада 2022 р.

**КАЛЕНДАРНИЙ ПЛАН**  
**Виконання магістерської кваліфікаційної роботи**

Тема: Інтелектуальна система розпізнавання та подолання перешкод  
транспортними засобами

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників МКР	17.10.2022	17.10.2022	Виконано
2	Отримання завдання на виконання МКР	07.11.2022	07.11.2022	Виконано
3	Складання календарного плану роботи на весь період виконання МКР	11.11.2022	11.11.2022	Виконано
4	Отримання завдання на переддипломну практику	21.11.2022	21.11.2022	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	28.11.2022	18.12.2022	Виконано
6	Розробка звіту з переддипломної практики	19.12.2022	20.12.2022	Виконано
7	Дипломне проектування	19.12.2022	03.02.2023	Виконано
8	Перший контроль виконання МКР	11.01.2023	11.01.2023	Виконано
9	Другий контроль виконання МКР	23.01.2023	23.01.2023	Виконано
10	Попередній захист МКР на засіданні комісії кафедри	03.02.2023	03.02.2023	Виконано
11	Доробка та остаточне оформлення МКР	03.02.2023	04.02.2023	Виконано
12	Подання МКР рецензенту	09.02.2023	09.02.2023	Виконано
13	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	15.02.2023	15.02.2023	Виконано
14	Захист МКР перед ЕК	22.02.2023	22.02.2023	Виконано

Розробив студент Миронюк О.М.

*(прізвище та ініціали)*

*(підпис)*

Керівник роботи канд. техн. наук, доц. Сіденко Є.В.

*(наук. ступінь, вчене звання, прізвище та ініціали)*

*(підпис)*

«11» листопада 2022 р.

## АНОТАЦІЯ

до магістерської кваліфікаційної роботи  
студента групи 601 ЧНУ ім. Петра Могили

**Миронюка Олександра Миколайовича**

на тему: **“ІНТЕЛЕКТУАЛЬНА СИСТЕМА РОЗПІЗНАВАННЯ ТА  
ПОДОЛАННЯ ПЕРЕШКОД ТРАНСПОРТНИМИ ЗАСОБАМИ”**

**Актуальність** даного дослідження полягає у недостатній навченості систем розпізнавання перешкод, що несе в собі небезпеку для учасників дорожнього руху. Багато дорожніх процесів, зокрема проїзд складного перехрестя, зміна смуги у переповненому авто потоці та визначення динамічного коридору є важкими процесами для сучасних систем розпізнавання і подолання перешкод транспортними засобами та небезпечними для людей.

**Об’єктом** дослідження є процеси руху та керування транспортними засобами.

**Предметом** дослідження є методи та засоби розпізнавання і подолання перешкод транспортними засобами.

**Метою** дослідження є розпізнавання та подолання перешкод транспортними засобами за рахунок розробки інтелектуальної системи і застосування нейронних мереж.

В результаті виконання роботи було досліджено чотири нейронних технології для навчання розпізнавання перешкод (LSTM, RNN, GRU та BRRN), проаналізовано їх особливість процесу розпізнавання, визначені основні їх переваги та недоліки, а також розроблено інтелектуальну систему, яка відображена у вебзастосунку.

Дана робота складається з шести розділів. Кожен розділ відповідно присвячений: аналізу предметної області, огляд і аналіз наявних аналогів та публікацій, технологіям, інструментам та методам, використаним у магістерській роботі, моделюванню і проектуванню системи планування і оптимізації маршрутів, аналізу отриманих результатів, охороні праці, методичній частині магістерської роботи. Загальний обсяг роботи – 123 сторінки. Магістерська кваліфікаційна робота містить два додатки, 121 рисунок, 4 таблиці і посилання на 50 літературних джерел.

**Ключові слова:** інтелектуальна система, вебдодаток, нейронна мережа, автоматизоване керування, JavaScript, машинне навчання.

## **ABSTRACT**

to the master's qualification work by the student of the group 601 of Petro Mohyla  
Black Sea National University

**Myroniuk Oleksandr**

### **“INTELLIGENT VEHICLE OBSTACLE RECOGNITION AND OVERCOME SYSTEM”**

A relevance of this study lies in insufficient training of obstacle recognition systems, which poses a danger to road users. Many traffic processes, in particular, driving through a complex intersection, changing lanes in a crowded traffic flow, and determining a dynamic corridor are difficult processes for modern systems of recognition and overcoming obstacles by vehicles and dangerous for people.

An object of research is the processes of movement and control of vehicles.

A subject of the research is the methods and means of recognizing and overcoming obstacles by vehicles.

A purpose of the study is to study is to recognize and overcome obstacles by vehicles through the development of an intelligent system and the use of neural networks.

As a result of the work, four neural technologies for training obstacle recognition (LSTM, RNN, GRU and BRRN) were investigated, their feature of the recognition process was analyzed, their main advantages and disadvantages were determined, and an intelligent system was developed, which is displayed in a web application.

This work consists of six sections. Each of them is devoted to: analysis of the subject area, review and analysis of existing analogues and publications, technologies, tools and methods used in the master's thesis, modeling and design of the route planning and optimization system, analysis of the obtained results, labor protection and life safety, methodological part of the master's work.

The overall scope of the work is 123 pages. Thesis contains 2 application, 121 figures, 4 tables and 50 sources in it.

**Key words:** intelligent system, web application, neural network, automated control, JavaScript, machine learning.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	9
ВСТУП.....	11
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ТА ПОСТАНОВКА ЗАДАЧІ .....	12
1.1 Опис предметної сфери .....	12
1.2 Огляд та аналіз наявних аналогів та публікацій .....	15
1.3 Постановка задачі .....	24
Висновки до розділу 1 .....	25
2 МЕТОДИ І ТЕХНОЛОГІЇ ДЛЯ РОЗПІЗНАВАННЯ ТА ПОДОЛАННЯ ПЕРЕШКОД ТРАНСПОРТНИМИ ЗАСОБАМИ .....	27
2.1 Технології створення вебзастосунків для вирішення поставленої задачі.....	27
2.2 Інструменти та методології для вирішення поставленої задачі .....	31
Висновки до розділу 2 .....	40
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ТА ПОДОЛАННЯ ПЕРЕШКОД ТРАНСПОРТНИМИ ЗАСОБАМИ .....	42
3.1 Механіка переміщення автомобіля .....	42
3.2 Створення дороги та смуг .....	45
3.3 Штучні датчики .....	47
3.4 Виявлення зіткнень .....	50
3.5 Імітація трафіку .....	52
3.6 Нейронна мережа .....	53
3.7 Розпаралелювання.....	57
3.8 Генетичний алгоритм.....	59

3.9 Результат роботи системи .....	60
3.10 Тестування роботи системи .....	60
Висновки до розділу 3 .....	62
<b>4 ДІАГРАМИ ВИКОРИСТАННЯ ТА ПЕРСПЕКТИВИ ВДОСКОНАЛЕННЯ</b>	<b>64</b>
4.1 Діаграми використання .....	64
4.2 Перспективи вдосконалення розробленої інтелектуальної системи .	71
<b>ВИСНОВКИ</b> .....	<b>75</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	<b>76</b>
<b>ДОДАТОК А Критерії бальної оцінки умов праці</b> .....	<b>80</b>
<b>ДОДАТОК Б Залежність категорії умов праці від величини інтегральної бальної оцінки</b> .....	<b>82</b>



## **ПЕРЕЛІК СКОРОЧЕНЬ**

AI	– Artificial intelligence
ML	– Machine Learning
API	– Application Programming Interface
RNN	– Recurrent neural network
BPTT	– Backpropagation through time
LSTM	– Long short-term memory
UML	– Unified Modeling Language

# **Пояснювальна записка**

**до магістерської кваліфікаційної роботи**

на тему:

## **«ІНТЕЛЕКТУАЛЬНА СИСТЕМА РОЗПІЗНАВАННЯ ТА ПОДОЛАННЯ ПЕРЕШКОД ТРАНСПОРТНИМИ ЗАСОБАМИ»**

Спеціальність 122 «Комп'ютерні науки»

**122 – МКР – 601. 21920103**

***Виконав студент 6-го курсу, групи 601***

***О. М. Миронюк***

*(підпис, ініціали та прізвище)*

«15» лютого 2023 р.

***Керівник: канд. техн. наук, доцент***

***Є. В. Сіденко***

*(підпис, ініціали та прізвище)*

«15» лютого 2023 р.

**Миколаїв – 2023**

## ВСТУП

У наш час більшість автомобільних концернів [1] сьогодні працюють зі створення автономного транспорту. Однак транспортні засоби зі справжнім автопілотом досі мають характер дослідницьких прототипів.

За оцінками різних компаній та організацій, повністю автоматизований транспорт займатиме значну частку серед пересувних засобів на дорогах світу вже у 2025-2050 роках. Це означає, що автомобілі будуть не тільки пересуватися самостійно, але й зможуть спілкуватися між собою за допомогою систем типу «Car-to-Car» [2], а також з навколишньою інфраструктурою [3], світлофорами, центрами дорожнього регулювання. Величезна кількість електронних систем та технологій необхідна для роботи системи автоматичного пілотування транспортного засобу [4-5], частина з яких вже сьогодні ефективно використовується у передових транспортних засобах.

Важливо відмітити той факт, що на теперішній час автомобілі стали майже невід'ємною частиною для людини. Вони надають масу переваг, таких як, незалежність від громадського транспорту, свобода переміщення та інші.

Актуальність даного дослідження полягає у недостатній навченості систем розпізнавання перешкод, що несе в собі небезпеку для учасників дорожнього руху. Багато процесів таких, як проїзд перетинів, зміна смуги у жвавому потоці та визначення динамічного коридору є важкими процесами для сучасних систем та небезпечними для людей.

Об'єктом дослідження є процеси руху та керування транспортними засобами.

Предметом дослідження є методи та засоби розпізнавання і подолання перешкод транспортними засобами.

Метою дослідження є розпізнавання та подолання перешкод транспортними засобами за рахунок розробки інтелектуальної системи і застосування нейронних мереж.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Опис предметної сфери

Головними обмеженнями автопілоту є погане розпізнавання об'єктів, високий відсоток аварій та смертельних випадків для пішоходів [6].

Дослідники, серед яких є і McKinsey [7-8] вважають, що до 2030 року у кращому разі 15% нових автомобілів будуть самоврядними. Але якщо виникнуть проблеми, то до 2040 року не більше ніж 5% автомобілів можна буде назвати автопілотованими [9]. Схема такого прогнозування продемонстрована на рис. 1.1.

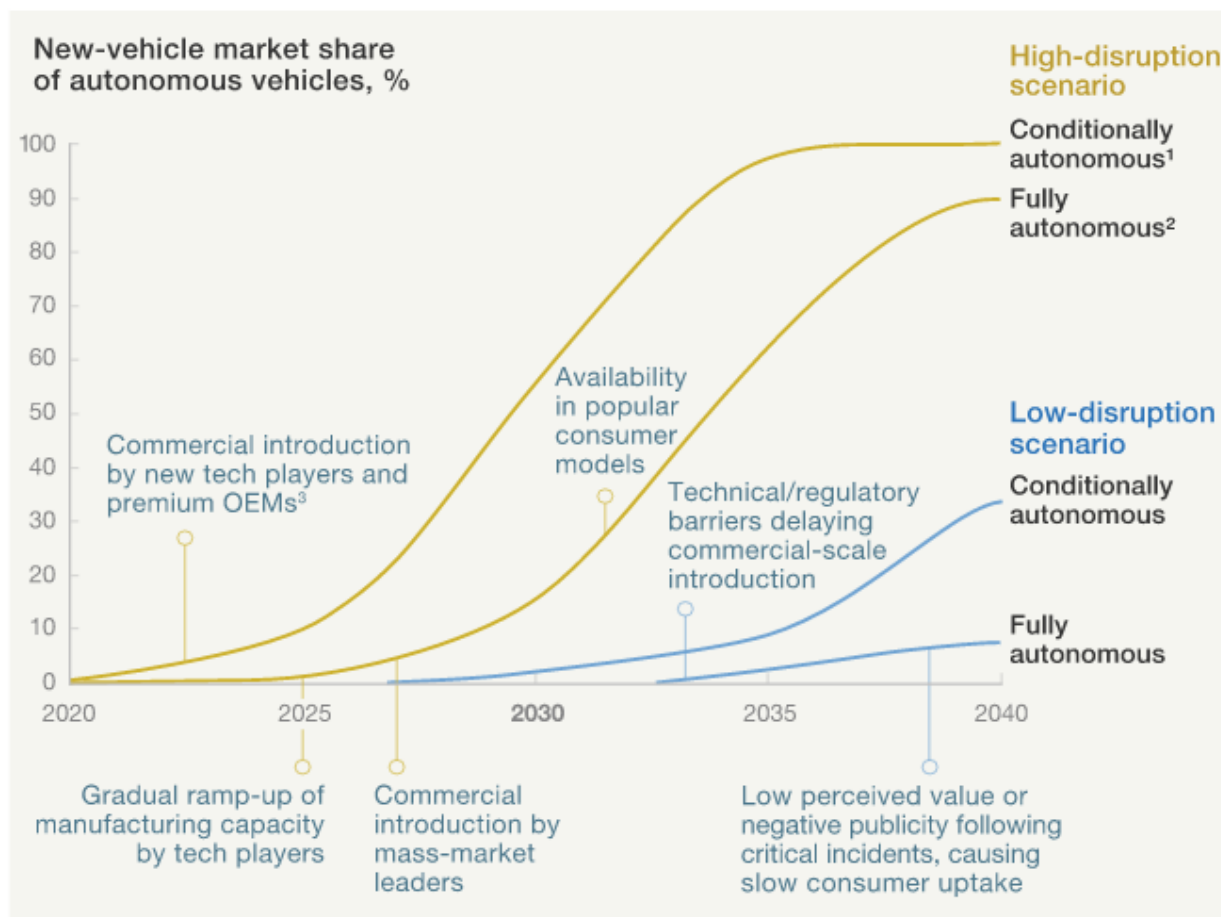


Рисунок 1.1 – Схема прогнозування автоматизації автомобілів

Серед проблем частіше виділяють наступні проблеми [10]: проблема розпізнавання об'єктів за допомогою датчиків, відсутність їх універсальності та ринкова проблема.

**Першою проблемою є датчики (рис. 1.2)**

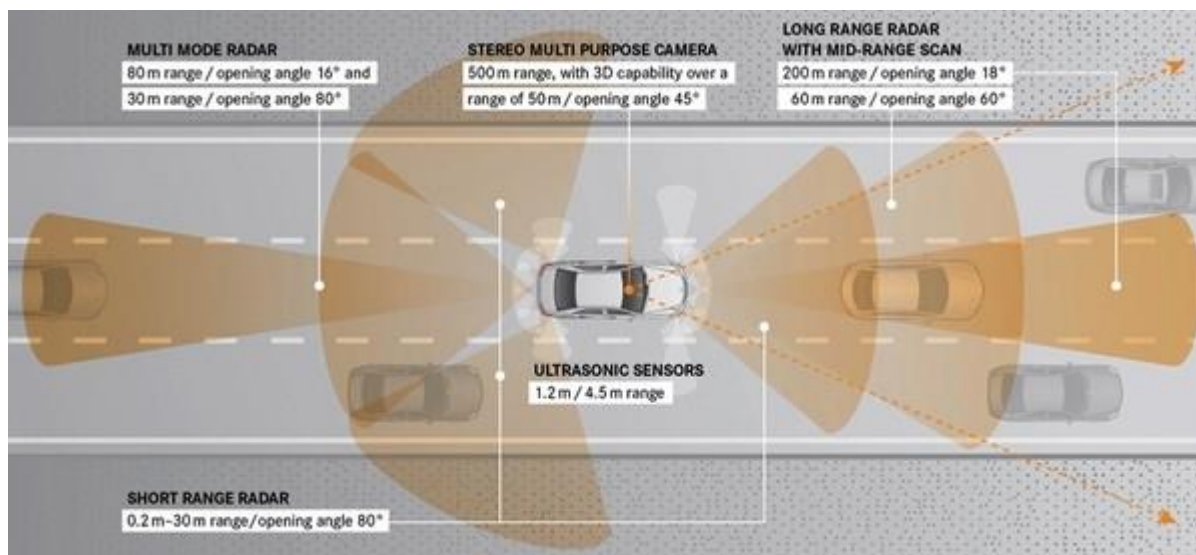


Рисунок 1.2 – Вигляд роботи датчиків

Головне гальмування розвитку автопілотів засноване на поганому розумінні автомобілем навколишнього простору. Інформації із сучасних датчиків [11] категорично недостатньо, також недостатньо навченості механізму прийняття рішень (нейромережі). Скануючі лідари, що стоять на більшості прототипів, такі як Google Car, дуже дорогі для серійних автомобілів і не дають необхідної інформації для впевненого руху повноти інформації [12].

Комп'ютер у будь-якому симуляторі буде майже завжди кращим за людину. Простою ілюстрацією відмінної роботи штучного інтелекту є: AI перемагає пілота на симуляторі повітряного бою в тій ситуації, коли він має повну інформацію про те, що відбувається. Якщо ж брати реальність, то на дорозі подібної переваги штучного інтелекту над людиною не відбудеться.

За існуючих датчиків (камери, радари, лідари тощо) повністю вирішити проблему самокерування взагалі неможливо.

Схоже питання з акумуляторами. Поки ємність акумуляторів не вийде підвищити в рази, а краще на порядок, електроніка, електрокари, що є наразі, теж зменшують темп розвитку автопілоту.

### **Друга проблема, відсутність універсальності.**

Прикладом даної проблеми є закон Парето, в автопілотах він теж використовується. 80% дорожніх ситуацій машини розуміють, 20%, що залишилися, плутаються та призводять до аварійних ситуацій [13-14].

З найпростішими завданнями автопілот функціонує [15], але не без помилок. Серед таких найпростіших завдань є автобанний автопілот, автопарковка, заїзд у гараж.

Розробникам залишилося реалізувати найважче: проїзд перетинів, зміна смуги у жвавому потоці та ще кілька десятків схожих сценаріїв. З деякими завданнями взагалі не зрозуміло, що робити. Наприклад, автомобілю складно заздалегідь зрозуміти, чи проїде він між двома перешкодами, що близько розташовані. І поки що неможливо зрозуміти свій динамічний коридор [16] (тобто межі простору, які автомобіль займає в динаміці), а без цього неможливо точно прорахувати дорожню ситуацію. Існуючі автопілоти настільки не універсальні, що вимагають кожен дорожній сценарій описувати окремо.

### **Проблема третя, ринкова.**

Автопілоту мало бути функціональною можливістю, для повноти потрібно стати ще і дешевшим. Автомобільної навігації понад двадцять років, але при ціні системи в кілька тисяч доларів вона досі ставиться менш ніж на 20% всіх нових автомобілів. Самоврядний функціонал ще дорожчий.

Дослідження Boston Consulting Group показали [16], що витратити на автопілот понад 5 тисяч доларів готові лише 17% покупців (американських покупців). Тобто щоб завоювати хоча б 15 відсотків світового ринку, системі автопілотування потрібно стати на порядок розумнішим і вдвічі дешевшим, ніж зараз. А це не виглядає близькою перспективою.

Але все ж таки помічників з елементами автопілотування (автопаркування, авторух у пробці, рух трасою з розміткою або навіть конвойні режими) в найближчі роки буде виготовлятися все більше. Автовиробники намагаються зробити їх дешевшими та масштабують ці функції, але поки що ще в недостатній кількості.

А у складних ситуаціях (ті самі 20 відсотків) управління на довгі роки залишиться на людині. Це тому що по-перше – машина не може адекватно оцінити те, що відбувається, і найближчим часом не навчиться.

Звикати до абревіатури ADAS, яка у найближчі роки стане такою ж звичною як стали ABS та ESP. Після цього варто чекати на активне держрегулювання в цій галузі, системи активної допомоги стануть обов'язковими в Європі та США.

І розуміти, що повноцінного автопілота поки що на ринку немає, і взагалі у них (поки що) непереборні проблеми.

## **1.2 Огляд та аналіз наявних аналогів та публікацій**

Всього в світі існує 6 рівнів автопілоту [17] (рис.1.3):

- рівень 0: відсутня автоматизація;
- рівень 1: допоміжна система (drive assist) може взяти на себе одну з функцій контролю керма, акселератора або гальм. Найпоширеніший приклад – адаптивний круїз-контроль;
- рівень 2: системи можуть керувати автомобілем у певних умовах – наприклад, на шосе з добре видимою розміткою. Водій завжди має бути готовий перехопити керування;
- рівень 3: тут допоміжні системи частково беруть на себе контроль оточення і водій може трохи розслабитися;
- рівень 4: у певних умовах і за наявності розмітки, що добре зчитується, автомобіль все робить сам;
- рівень 5: повна автоматизація в будь-яких умовах водій може займатися своїми справами і навіть спати.

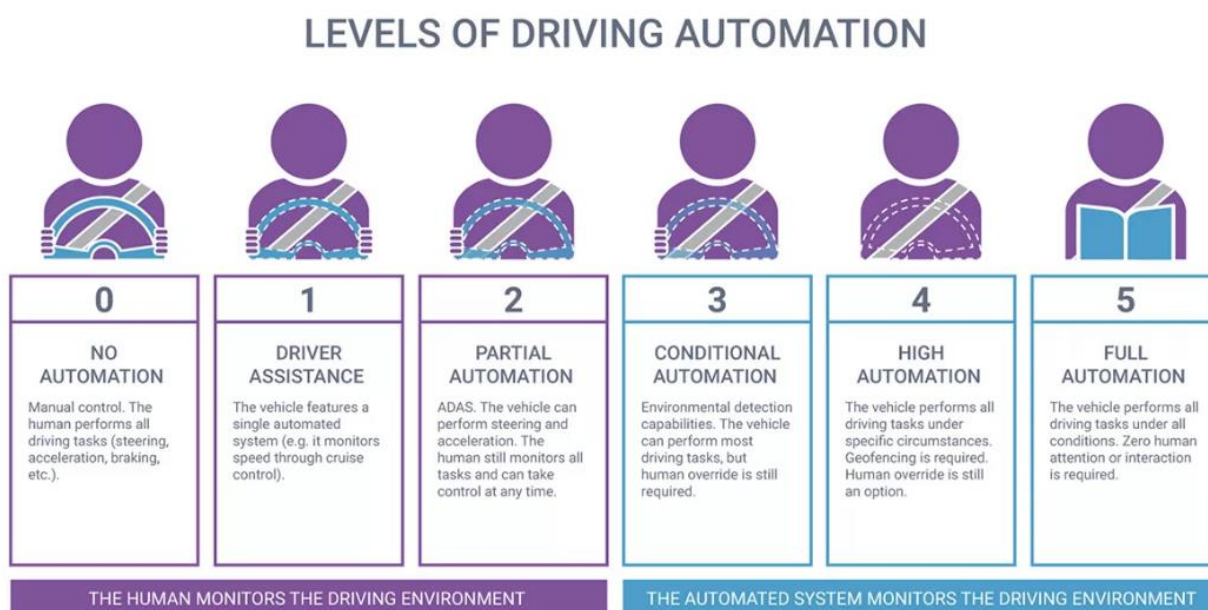


Рисунок 1.3 – Рівні автопілоту

В наш час існують наступні найпопулярніші компанії [18], які створили систему автопілоту, влаштовану в автомобілі:

- Mercedes-Benz (Drive Pilot) [19];
- Tesla Inc (Autopilot) [20];
- Google/Waymo (Waymo Driver) [21];
- Audi (Traffic Jam Pilot) [22];
- Baidu (Apollo) [23];
- General Motors (Super Cruise) [24];
- Toyota (Advanced Drive) [25];
- Wepod (спільна розробка з NVIDIA) [26].

На сьогодні найкращий автопілот, котрий отримав ліцензію має лише 3 рівень, він належить компанії Mercedes-Benz.

Нижче будуть описані автопілоти Tesla та Mercedes-Benz. У останніх автомобільних компаніях автопілоти мають велику схожість між собою та мають гірші можливості у порівнянні з Tesla чи Mercedes-Benz.

#### **Tesla Inc.**



Автомобілі Tesla постачаються з вдосконаленим апаратним забезпеченням, здатним забезпечувати функції автопілота та повні можливості самостійного водіння – завдяки оновленням програмного забезпечення, призначеним для покращення функціональності з часом.

### **Розширене покриття сенсора.**

Вісім камер і потужна обробка зору забезпечують 360 градусів видимості на відстані до 250 метрів (рис. 1.4).

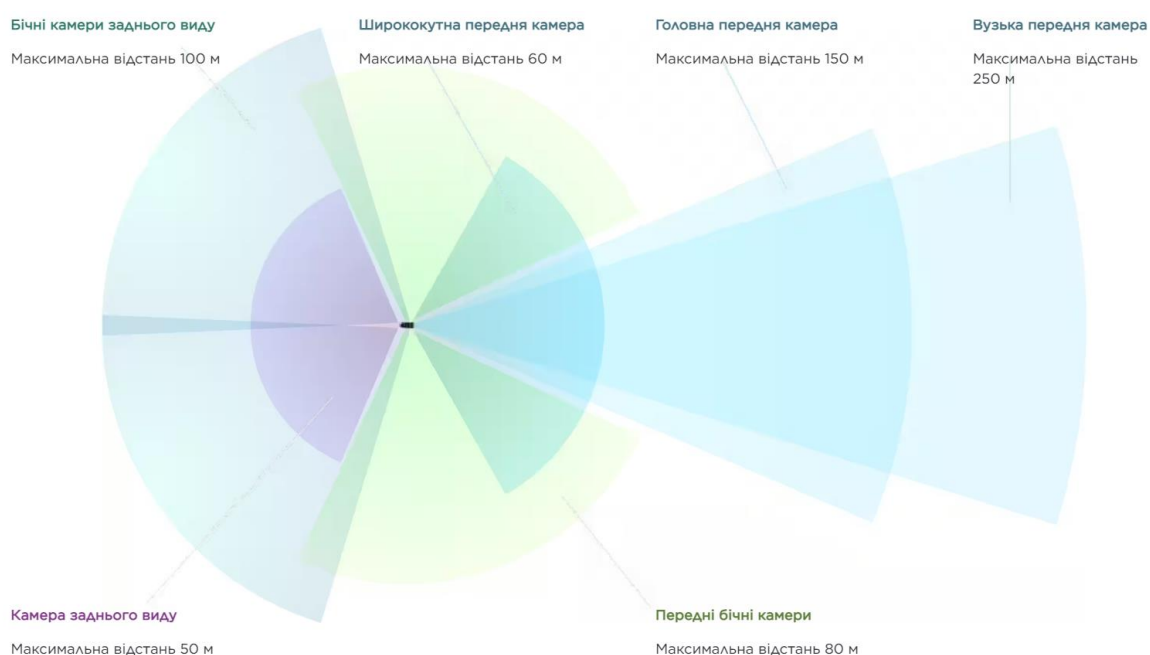


Рисунок 1.4 – Покриття сенсора автомобіля Tesla

### **Потужність обробки.**

Щоб зрозуміти всю дорожню інформацію, бортовий комп'ютер Hardware 3 обробляє дані. Цей комп'ютер керує нейронною мережею, розробленою Tesla, яка є основою для того, як навчається та розвивається автопілот. Ця система забезпечує огляд світу, до якого водій не може отримати доступ, бачить у всіх напрямках одночасно та на довжинах хвиль, які виходять далеко за межі людських почуттів.

Щоб використовувати такий потужний набір камер, кожен автомобіль Tesla має потужний набір інструментів обробки зображення, побудований на основі

глибокої нейронної мережі, Tesla Vision деконструює середовище автомобіля з більшою надійністю, ніж ті, які можна досягти за допомогою класичних методів обробки зору.

### **Автопілот.**

Розширені функції безпеки та зручності Autopilot розроблені, щоб допомогти у найважчих частинах водіння. Autopilot представляє нові функції та покращує наявні функції, щоб з часом зробити Tesla безпечнішою та продуктивнішою.

Автопілот дозволяє автомобілю автоматично керувати, прискорюватися та гальмувати в межах своєї смуги руху.

Поточні функції автопілота вимагають активного контролю водія і не роблять транспортний засіб автономним.

Приклад роботи Tesla Autopilot зображено на рис. 1.5.

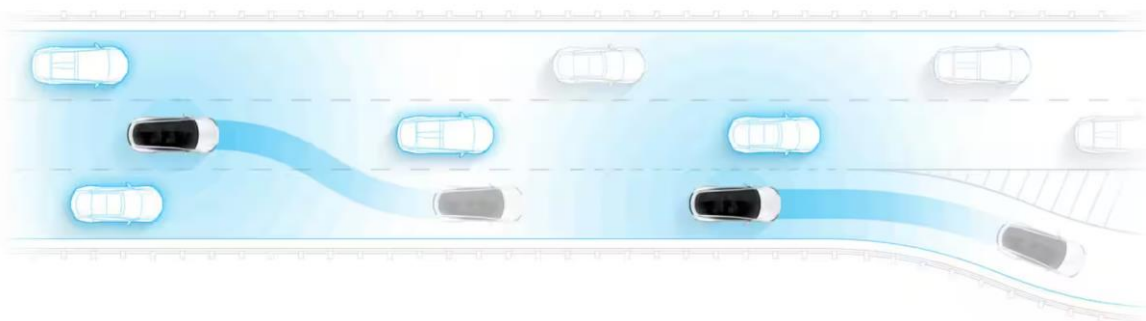


Рисунок 1.5 – Робота Tesla Autopilot

Навігація на автопілоті пропонує змінити смугу руху, щоб оптимізувати маршрут, і вносить корективи, щоб обмежити можливості застрягти за повільними автомобілями чи вантажівками. Якщо активовано, Navigate on Autopilot також автоматично спрямовуватиме автомобіль до розв'язок і з'їздів на шосе залежно від місця призначення.

### **Mercedes-Benz.**

Нова система високоавтоматизованого керування Drive Pilot є наступним рівнем розвитку системи Intelligent Drive. Система сама підтримує безпечну

відстань до інших учасників руху. У випадках, коли стає необхідним втручання людини, Drive Pilot видає звуковий сигнал.

В окремих ділянках руху швидкість може бути обмежена законом на позначці 60 км/год – у таких зонах Drive Pilot враховуватиме швидкісні обмеження. Для керування системою можна використовувати відповідні елементи на кермі. При активації система миттєво приймається регулювати швидкість та відстань до решти автомобілів, впевнено спрямовуючи автомобіль по вибраній смузі. Система є адаптивною та враховуватиме будь-які дорожні знаки, що відбуваються на дорозі попереду події та сам характер маршруту. І навіть у несподіваних дорожніх ситуаціях Drive Pilot готова упевнено реагувати. Це можуть бути маневри ухилення у межах смуги руху або використання гальм. В окремих ситуаціях, коли система вважає, що потрібне залучення водія, з'явиться відповідний звуковий сигнал.

Система Drive Pilot дозволяє рухатися в режимі автопілота 3-го рівня, тобто більшу частину часу система буде керувати автомобілем самостійно і лише в складних ситуаціях закликати водія взяти керування на себе. Такий випадок зображений на рис. 1.6.

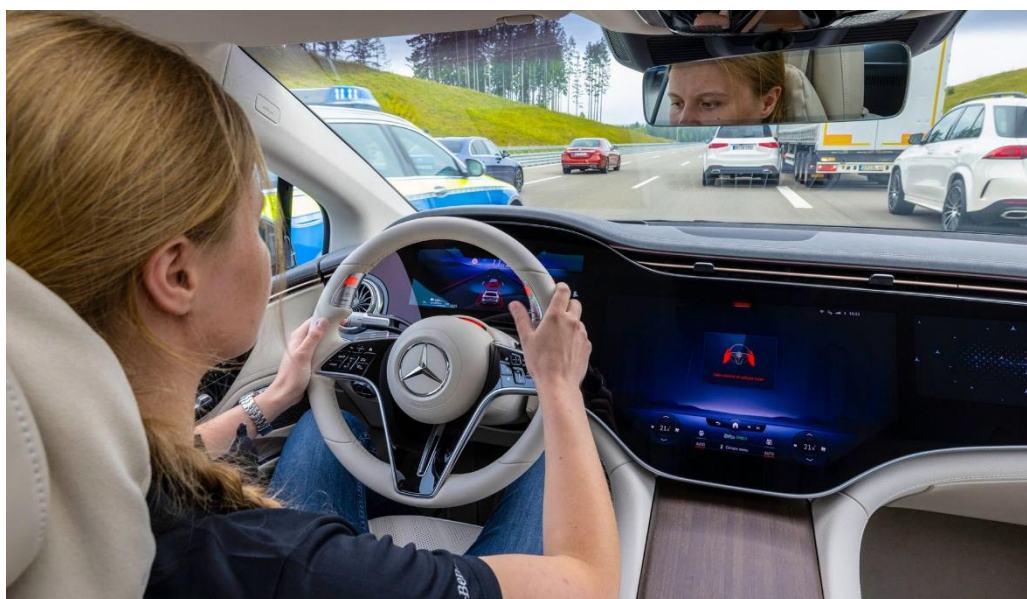


Рисунок 1.6 – Закликання водія взяти керування на себе

Система автопілота 3 рівня від Mercedes-Benz у своїй роботі спирається на датчики, які є частиною деяких систем допомоги водієві, вбудованих за замовчуванням. До них додаються і кілька унікальних датчиків, які Mercedes-Benz вважає за необхідне для безпечного та автоматизованого водіння. До них відносяться датчик LiDAR, додаткова камера у задньому склі, мікрофони. Також Drive Pilot використовує інформацію про геометрію дороги, властивості маршруту, дорожні знаки та особливі дорожні події (наприклад, аварії або будівельні майданчики) з карти. Положення автомобіля визначається за допомогою високоточної системи позиціонування, яка виходить далеко за межі звичайних систем GPS.

Крім того, з опцією DRIVE PILOT, S-Клас отримує резервні системи рульового управління та гальмування, а також резервну бортову мережу, щоб їм можна було маневрувати навіть у разі відмови однієї з цих систем, а також для забезпечення безпечного перемикання керування на водія. Продуктивний центральний блок управління реалізує складні програмні функції, необхідні високоавтоматизованого водіння. Наприклад, при обробці зображень використовуються технології в галузі штучного інтелекту. У межах сучасної архітектури безпеки всі алгоритми розраховуються двічі.

### **Google/Waymo.**

Waymo Driver – це автономний водій, який проїхав мільйони миль через незліченну кількість ситуацій на дорогах загального користування та ще мільярди в моделюванні, зібрано велику кількість даних і безцінні уроки, щоб розвинути технологію автономного водіння далі.

Перш ніж водій Waymo почне працювати в новій місцевості, спершу наноситься неймовірно детальна карта території, від маркерів смуг до знаків зупинки до бордюрів і пішохідних переходів. Потім, замість того, щоб покладатися виключно на зовнішні дані, такі як GPS, які можуть втрачати сигнал, Waymo Driver

використовує ці високодетальні спеціальні карти, зіставлені з даними датчиків у реальному часі, щоб завжди визначати своє точне місцезнаходження на дорозі.

Система сприйняття водія Waymo отримує складні дані, зібрані з передового набору датчиків, і розшифровує те, що відбувається навколо, за допомогою таких технологій, як машинне навчання – від пішоходів до велосипедистів, транспортних засобів до будівництва тощо. Waymo Driver також реагує на знаки та сигнали, як-от кольори світлофора та знаки тимчасової зупинки.

Ситуації водіння можуть включати сотні об'єктів, кожен зі своєю унікальною поведінкою та намірами. Waymo Driver використовує інформацію, яку він збирає в режимі реального часу. Він розуміє, чому автомобіль рухається інакше, ніж велосипедист, пішохід чи інший об'єкт, а потім миттєво передбачає численні можливі шляхи, якими можуть рухатися інші учасники дорожнього руху.

Waymo Driver збирає всю цю інформацію – від своїх дуже детальних карт до об'єктів навколо та куди вони можуть рухатися – і планує найкращу дію чи маршрут. Він миттєво визначає точну траєкторію, швидкість, смугу руху та маневри керування, необхідні для безпечної поведінки під час подорожі.

За останнє десятиліття Waymo розробила єдину інтегровану систему датчиків і обчислювальних пристроїв, призначених для спільної роботи, щоб надати Waymo Driver повне уявлення про навколишній світ. Незалежно від того, день чи ніч, поблизу чи далеко, Waymo Driver створений, щоб чітко бачити, що відбувається.

Лідар, або Light Detection and Ranging, створює тривимірну картину оточення автомобіля. Лідарні датчики розташовані навколо автомобіля, щоб надсилати мільйони лазерних імпульсів у всіх напрямках, а потім вимірювати, скільки часу потрібно, щоб вони відскочили від об'єктів. Незалежно від часу доби система лідарів дає Waymo Driver змогу бачити все навколо з висоти пташиного польоту (рис. 1.7).



Рисунок 1.7 – Лідар

Камери дають водієві Waymo одночасний огляд на 360° навколо автомобіля. Вони розроблені з високим динамічним діапазоном і термічною стабільністю, щоб бачити як за денного світла, так і за умов слабого освітлення, а також працювати зі складнішими середовищами. Вони можуть помітити світлофори, будівельні зони та інші об'єкти сцени навіть на відстані сотень метрів. На автомобілях Chrysler Pacifica є 19 камер, а на Jaguar I-PACE – 29 камер (рис. 1.8).

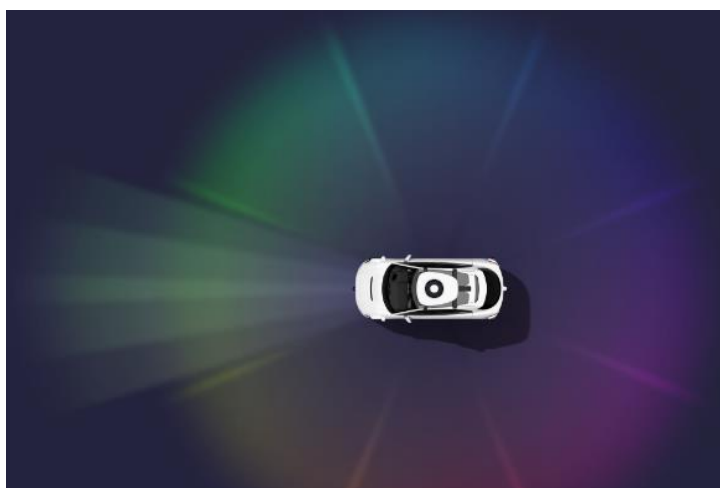


Рисунок 1.8 – Камери

Радар (рис. 1.9) використовує хвилі міліметрової частоти, щоб надати драйверу Waymo такі важливі деталі, як відстань і швидкість об'єкта. Радар ефективний в дощ, туман і сніг.



Рисунок 1.9 – Радар

Як «мозок» водія Waymo, бортовий комп'ютер поєднує в собі найновіші центральні та графічні процесори серверного класу. Він отримує інформацію, надану десятками датчиків автомобіля, ідентифікує різні об'єкти (наприклад, інші автомобілі та пішоходів) і планує безпечний шлях, маршрут до місця призначення – все в режимі реального часу.

Тестування є основою всього. Ретельно перевіряються Waymo Driver у сценаріях, які є більш складними, ніж ті, з якими стикаються на дорозі, щоб підтвердити, що він відповідає внутрішнім критеріям готовності до безпеки, і дотримуємося всіх відповідних законів штату та федеральних правил для тестування.

Waymo One, служба замовлення поїздок, щодня обслуговує пасажирів у метро Фенікс і Сан-Франциско, допомагаючи їм дістатися, куди вони прямують.

Waymo Via спеціалізується на транспортуванні комерційних вантажів. Рішення для автономного водіння розроблено для підвищення безпеки та



ефективності, тоді як наше рішення для місцевої доставки покликане революціонізувати досвід «останньої милі».

Waymo One та Waymo Via зображені на рис. 1.10.



Рисунок 1.10 – Waymo One та Waymo Via

### 1.3 Постановка задачі

Дослідивши можливості існуючих аналогічних систем, було виявлено обмеження роботи даних систем.

Обмеженнями систем є їх погане розпізнавання об'єктів, високий відсоток аварій та смертельних випадків для пішоходів. Інформації з сучасних датчиків та вдосконалення механізму прийняття рішень (нейромережі) категорично недостатньо. Дуже складним є проїзд перетинів, переміщення зі смуги в іншу смугу в жвавому потоці та ще багато прикладів. Неможливо зрозуміти свій динамічний коридор (тобто межі простору, які автомобіль займає в динаміці), а без цього неможливо правильно прорахувати дорожню ситуацію.

Розглянувши аналогічні інтелектуальні системи, можна зробити висновки, як буде побудована нейронна мережа та які матиме функції. Серед них кращий збір інформації на дорозі датчиками шляхом їх вдосконалення, прорахованість усіх можливих випадків уникнення перешкод, процес моделювання від найпростіших до найскладніших ситуацій для навчання нейронної мережі.

Об'єктом є процеси руху та керування транспортними засобами.



Предметом є методи та засоби розпізнавання і подолання перешкод транспортними засобами.

Метою даної роботи є розробка інтелектуальної системи для уникнення перешкод автоматичним способом шляхом використання нейронної мережі.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- проаналізувати сучасний стан задачі розпізнавання та подолання перешкод транспортними засобами;
- реалізувати обраний вид нейронної мережі, який дозволить збирати дані, передавати їх між собою, розраховувати результат та керувати автомобілем для уникнення перешкод;
- розробити інтелектуальну систему для розпізнавання та подолання перешкод транспортними засобами у вигляді вебзастосунку;
- протестувати роботу розробленої системи, визначити ефективність та швидкість проходження змодельованих ситуацій шляхом зміни параметрів мутації та кількості об'єктів для розрахунку усіх можливих варіантів уникнення різноманітних перешкод.

## **Висновки до розділу 1**

Дослідивши проблеми автопілоту, можна виділити обмеження автопілоту, серед яких є погане розпізнавання об'єктів, високий відсоток аварій та смертельних випадків для пішоходів.

Це зумовлено рядом обмежень систем:

- зчитування датчиками інформації на дорогах;
- відсутність універсальності;
- ринкова проблема.

Всього в світі існує 6 рівнів автопілоту, але на даний момент найкращий автопілот, котрий отримав ліцензію має лише 3 рівень, він належить компанії Mercedes-Benz.

Серед систем аналогів розглянуто найбільші компанії, так як Tesla Inc та Mercedes-Benz. Вони мають передові технології, влаштовані в їх системи автопілотування. Серед них є автоматичне керування автомобілем за допомогою нейронних мереж шляхом зчитування інформації з лідарів. При поганому розумінні дорожньої ситуації автомобіль подає спеціальний звуковий сигнал. Присутня велика кількість лідарів по всьому автомобілю з різною характеристикою та застосуванням.

Розглянувши аналогічні інтелектуальні системи, можна зробити висновки, як буде побудована нейронна мережа та які матиме функції. Серед них кращий збір інформації на дорозі датчиками шляхом їх вдосконалення, прорахованість усіх можливих випадків уникнення перешкод, процес моделювання від найпростіших до найскладніших ситуацій для навчання нейронної мережі.

Для проектування інтелектуальної системи визначено задачі.

## 2 МЕТОДИ І ТЕХНОЛОГІЇ ДЛЯ РОЗПІЗНАВАННЯ ТА ПОДОЛАННЯ ПЕРЕШКОД ТРАНСПОРТНИМИ ЗАСОБАМИ

### 2.1 Технології створення вебзастосунків для вирішення поставленої задачі

Існує дві основні категорії кодування, сценаріїв і програмування для створення вебзастосунків [27]:

**Категорія 1. Сценарії/кодування на стороні клієнта.** Сценарії на стороні клієнта – це тип коду, який виконується або інтерпретується браузером.

Сценарії на стороні клієнта зазвичай доступні для перегляду будь-якому відвідувачу застосунків.

Нижче наведено деякі поширені технології сценаріїв на стороні клієнта:

- HTML (мова розмітки гіпертексту);
- CSS (каскадні таблиці стилів);
- JavaScript;
- Ajax (асинхронний JavaScript і XML);
- jQuery, MooTools, Dojo Toolkit (бібліотеки JavaScript Framework – зазвичай використовується в розробці Ajax).

**Категорія 2. Сценарії/кодування на стороні сервера.** Сценарії на стороні сервера – це тип коду, який виконується або інтерпретується вебсервером.

Сценарії на стороні сервера недоступні для перегляду та доступу для будь-якого відвідувача чи широкого загалу.

Нижче наведено поширені технології серверних сценаріїв:

- PHP (дуже поширена мова сценаріїв на стороні сервера – Linux / Unix на основі відкритого коду – безкоштовне розповсюдження, зазвичай поєднується з базою даних MySQL);
- Zend Framework (об'єктно-орієнтована платформа вебзастосунків PHP);
- ASP (мова сценаріїв вебсервера Microsoft (IIS));

- ASP.NET (Microsoft's Web Application Framework – спадкоємець ASP);
- ColdFusion (платформа вебзастосунків Adobe);
- Ruby on Rails (Framework Web Application Framework для програмування Ruby – безкоштовне розповсюдження);
- Python (мова програмування високого рівня загального призначення та мова сценаріїв на стороні сервера – безкоштовне розповсюдження).

### **Бібліотеки програм.**

Бібліотеки програм [28] – це набір часто використовуваних функцій, класів або підпрограм, які спрощують розробку та обслуговування, дозволяючи розробникам легко додавати або редагувати функціональні можливості до каркасної або модульної програми.

### **Фреймворки вебзастосунків.**

Платформи вебзастосунків – це набори програмних бібліотек, компонентів і інструментів, організованих в архітектурну систему, що дозволяє розробникам створювати та підтримувати складні проекти вебзастосунків, використовуючи швидкий і ефективний підхід.

Платформи вебзастосунків розроблені для оптимізації програмування та сприяння повторному використанню коду шляхом визначення організації та структури папок, документації, інструкцій і бібліотек (кодів для загальних функцій і класів, які можна багаторазово використовувати).

### **Фреймворки вебзастосунків – переваги та переваги.**

Дії та логіка програми [29] відокремлені від файлів HTML, CSS і дизайну. Це допомагає дизайнерам (без досвіду програмування) мати можливість редагувати інтерфейс і вносити зміни в дизайн без допомоги програміста.

Збірки базуються на модулі, бібліотеках та інструментах, що дозволяє програмістам легко ділитися бібліотеками та впроваджувати складні функціональні можливості та функції швидким і ефективним способом.

Структура допомагає створювати найкращі методи кодування з узгодженою логікою та стандартами кодування, а також надає іншим розробникам можливість ознайомитися з кодом за короткий час.

### **Інструкції з кодування, стандарти та конвенції.**

Інструкції з кодування – це набори правил і стандартів, які використовуються в програмуванні проекту вебзастосунку.

Ці правила та стандарти застосовуються до логіки кодування, структури та імен папок, імен файлів, організації файлів, форматування та відступів, операторів, класів і функцій, а також угод про іменування. Ці правила також вимагають написання чітких коментарів і надання документації.

### **Важливі переваги використання вказівок із кодування:**

- створює найкраще середовище для роботи кількох програмістів над одним проектом;
- забезпечує легкість обслуговування та керування версіями;
- забезпечує кращу читабельність і розуміння вихідного коду;
- гарантує, що інші розробники зможуть зрозуміти та ознайомитися з кодом за короткий час.

### **Модель життєвого циклу вебзастосунків.**

Життєвий цикл вебзастосунку [30] – це процес розробки вебзастосунку та залучення кількох команд, які беруть участь у процесі розробки. Кожна організація може запропонувати свій унікальний стиль діяльності.

Деякі компанії дотримуються певної стандартної моделі, такої як SDLC (Життєвий цикл розробки системи) або Agile Software Development Model.

- SDLC – це традиційний процес розробки програмного забезпечення або вебзастосунків, який включає дослідження для виявлення та визначення вимог до застосунків, аналіз інформації, архітектурний дизайн і план специфікацій, залучення команди, програмування, тестування та виправлення помилок, тестування системи, впровадження та обслуговування;

– гнучка розробка програмного забезпечення/вебзастосунків – це ітеративний процес розробки та практики процесу розробки, які зосереджуються на співпраці залучених людей і забезпечують кращу процедуру, що дозволяє переглядати та розвивати вимоги до вебзастосунків. Гнучка методологія включає дослідження, аналіз, управління проектами, дизайн, програмування, впровадження, часте тестування, адаптацію та обслуговування.

### **Процес розробки вебзастосунків.**

Процес розробки вебзастосунків організовує практичну процедуру та підхід до розробки застосунків.

Наведений нижче перелік процедур і запропонованих документів надає гарний опис життєвого циклу та процесу вебзастосунку:

- документ дорожньої карти: визначення вебзастосунку, мети, цілей і напрямку;
- дослідження та визначення сфери аудиторії та безпеки документів;
- створення функціональних специфікацій або документа з описом функцій;
- командна співпраця та документ про управління проектом;
- вибір технології, технічні специфікації, ілюстративна діаграма архітектури та структури вебзастосунку, методологія розробки, керування версіями, резервне копіювання, оновлення, розширення та розширення документа щодо планування, вибір серверного апаратного/програмного забезпечення;
- аналіз і вибір сторонніх постачальників (обліковий запис продавця та платіжний шлюз, SSL-сертифікат, керований сервер/постачальник суміжних серверів, центри виконання, програмне забезпечення для аналізу відвідувачів веб-сайту, сторонні системи перевірки тощо);
- візуальний посібник із застосування, макет дизайну, дизайн інтерфейсу, каркас;
- проектування структури бази даних і розробка вебзастосунків;

- тестування: гарантія якості, сумісність із кількома браузерами, безпека, продуктивність;
- навантаження та стрес-тестування, зручність використання;
- технічне обслуговування.

### **Тестування вебзастосунків**

Тестування є важливою частиною процесу розробки вебзастосунків. Інколи тестування забирає більше робочої сили та часу, ніж сама розробка.

Нижче наведено деякі з найпоширеніших тестів, необхідних для будь-якого процесу розробки вебзастосунків:

- гарантія якості та тестування помилок;
- сумісність із кількома браузерами;
- безпека програми;
- продуктивність – тестування на навантаження та стрес.

## **2.2 Інструменти та методології для вирішення поставленої задачі**

Для розробки системи використано мову JavaScript [31-32], рекурентну нейронну мережу (RNN) та Microsoft Visual Studio.

### **JavaScript.**

Завдяки новим бібліотекам розробники JavaScript тепер можуть створювати програми ML без Python або R. Таким чином JavaScript може допомогти розробникам перенести ML у браузер і в Інтернет.

Завдяки бібліотекам ML розробники JavaScript можуть додавати штучний інтелект до вебзастосунків.

Більшість програм машинного навчання сьогодні використовують R або Python.

Але JavaScript має велике майбутнє як мова машинного навчання, і він навіть має деякі переваги:

- JavaScript більш відомий. Ним можуть користуватися всі розробники;

- вбудована безпека. JavaScript не може отримати доступ до ваших файлів;
- JavaScript швидший за Python;
- сучасний JavaScript компілюється в машинний код;
- сучасний JavaScript може використовувати апаратне прискорення.

### **WebGL API.**

WebGL [33] – це API JavaScript для відтворення двовимірної та тривимірної графіки в будь-якому браузері.

WebGL може працювати як на вбудованих, так і на автономних графічних картах будь-якого ПК.

WebGL приносить 3D-графіку у веб-переглядач. Основні постачальники браузерів Apple (Safari), Google (Chrome), Microsoft (Edge) і Mozilla (Firefox) є членами робочої групи WebGL.

Прикладом є рис. 2.1.

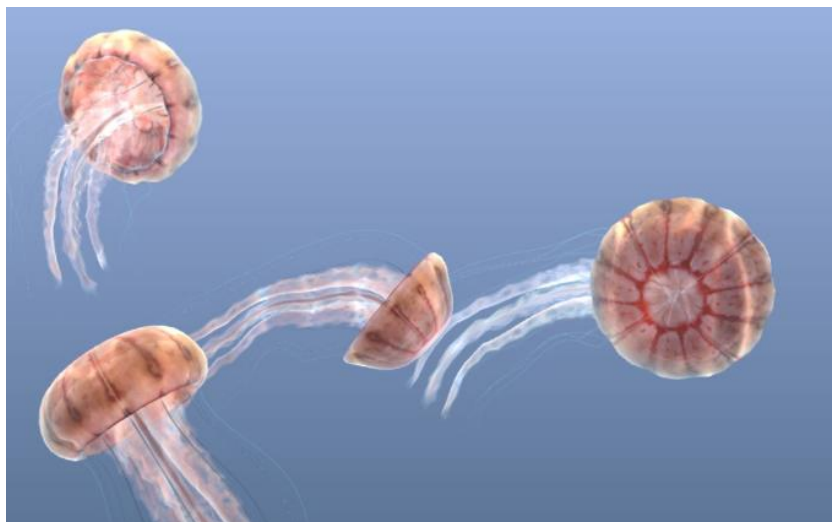


Рисунок 2.1 – 3D-графіка за допомогою WebGL

Машинне навчання може бути важким для математики. Природа нейронних мереж суворо технічна, і жаргон, який поєднується з нею, як правило, відлякує людей.



Тут на допомогу може прийти JavaScript. Нам потрібні прості для розуміння програмні API, щоб спростити процес створення та навчання нейронних мереж.

### **Бібліотеки в JavaScript для побудови нейронних мереж.**

#### **Brain.js.**

Brain.js [34] – це бібліотека JavaScript, яка полегшує розуміння нейронних мереж, оскільки приховує складність математики.

#### **TensorFlow Playground**

TensorFlow Playground [35] – це веб-програма, написана на d3.js.

З TensorFlow Playground можна дізнатися про нейронні мережі (NN) без математики. Є можливість у веб-браузері створити нейронну мережу та побачити результат. TensorFlow.js раніше називався Tf.js і Deeplearn.js.

#### **ml5.js**

ml5.js [36] намагається зробити машинне навчання доступнішим для ширшої аудиторії. Команда ml5 працює над зручнішим використанням функцій машинного навчання.

#### **Нейронна мережа RNN.**

Повторювана нейронна мережа (RNN) [37-38] – це тип штучної нейронної мережі, яка використовує послідовні дані або дані часових рядів.

Ці алгоритми глибокого навчання [39-40] зазвичай використовуються для порядкових або часових проблем [41], таких як переклад мови, обробка природної мови (nlp), розпізнавання мовлення та підписи до зображень; вони включені в такі популярні програми, як Siri, голосовий пошук і Google Translate. Подібно до нейронних мереж прямого зв'язку та згорткових нейронних мереж (CNN) [42], рекурентні нейронні мережі використовують навчальні дані для навчання. Вони відрізняються своєю «пам'яттю», оскільки вони беруть інформацію з попередніх вхідних даних, щоб впливати на поточні вхідні та вихідні дані. Хоча традиційні глибокі нейронні мережі припускають, що входи та виходи незалежні один від

одного, вихід рекурентних нейронних мереж залежить від попередніх елементів у послідовності.

Порівняння рекурентних нейронних мереж (ліворуч) і нейронних мереж прямого зв'язку (праворуч) зображено на рис. 2.2.

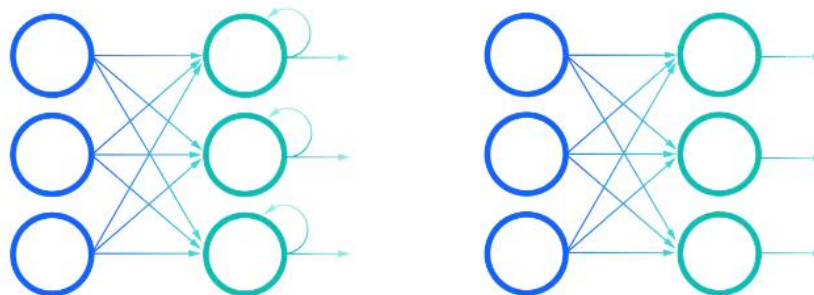


Рисунок 2.2 – Повторювана нейронна мережа (RNN) та пряма нейронна мережа

Дивлячись на зображення нижче (рис. 2.3), «згорнуте» зображення RNN представляє всю нейронну мережу, точніше, всю передбачувану фразу. «Розгорнутий» візуал представляє окремі шари або часові кроки нейронної мережі. Кожен шар відповідає одному слову у цій фразі, наприклад «погода». Попередні вхідні дані, такі як «відчуття» та «під», будуть представлені як прихований стан на третьому часовому кроці для прогнозування виходу в послідовності «the».

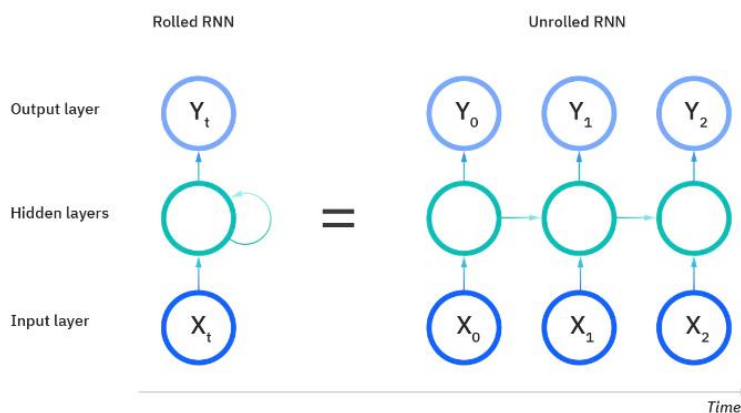


Рисунок 2.3 – «згорнуте» зображення RNN

Іншою відмінною характеристикою рекурентних мереж є те, що вони спільно використовують параметри на кожному рівні мережі. У той час як мережі прямого зв'язку мають різні ваги для кожного вузла, рекурентні нейронні мережі мають однаковий параметр ваги на кожному рівні мережі. Тим не менш, ці ваги все ще коригуються за допомогою процесів зворотного поширення та градієнтного спуску, щоб полегшити навчання з підкріпленням.

Повторювані нейронні мережі використовують алгоритм зворотного поширення в часі (ВРТТ) для визначення градієнтів, який дещо відрізняється від традиційного зворотного поширення, оскільки він специфічний для даних послідовності. Принципи ВРТТ такі ж, як і традиційного зворотного поширення, коли модель тренується, обчислюючи помилки від вихідного до вхідного рівня. Ці розрахунки дозволяють нам правильно налаштувати та підігнати параметри моделі. ВРТТ відрізняється від традиційного підходу тим, що ВРТТ підсумовує помилки на кожному кроці часу, тоді як упереджувальних мережах не потрібно підсумовувати помилки, оскільки вони не поділяють параметри на кожному рівні.

Через цей процес RNN, як правило, стикаються з двома проблемами, відомими як вибухові градієнти та зникнення градієнтів. Ці проблеми визначаються розміром градієнта, який є нахилом функції втрат уздовж кривої помилок. Коли градієнт занадто малий, він продовжує зменшуватися, оновлюючи вагові параметри, поки вони не стають незначними, тобто 0. Коли це відбувається, алгоритм більше не навчається. Вибухові градієнти виникають, коли градієнт занадто великий, що створює нестабільну модель. У цьому випадку ваги моделі виростуть занадто великими, і в кінцевому підсумку вони будуть представлені як NaN. Одним із рішень цих проблем є зменшення кількості прихованих шарів у нейронній мережі, усунення частини складності в моделі RNN.

У RNN результат обчислень на кожному етапі використовується як вихідні дані для наступного. Завдяки цьому вона може обробляти серії подій у часі або

послідовності для отримання результату обчислень. Це особливо корисно, коли автомобілю потрібні дані про свій динамічний простір.

### Типи рекурентних нейронних мереж.

Мережі прямого зв'язку відображають один вхід на один вихід, і хоча візуалізувалися рекурентні нейронні мережі таким чином на наведених вище діаграмах, вони насправді не мають цього обмеження. Натомість їхні вхідні та вихідні дані можуть відрізнятися за довжиною, а різні типи RNN використовуються для різних випадків використання, таких як створення музики, класифікація настроїв і машинний переклад. Різні типи RNN зазвичай виражаються за допомогою таких діаграм:

- один до одного (рис. 2.4);



Рисунок 2.4 – Один до одного

- один до багатьох (рис. 2.5);

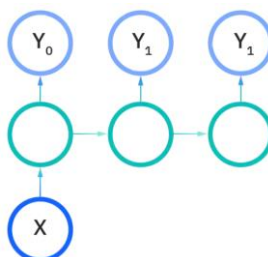


Рисунок 2.5 – Один до багатьох

- багато до одного (рис. 2.6);

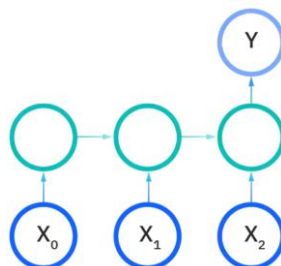


Рисунок 2.6 – Багато до одного

– багато до багатьох (рис. 2.7).

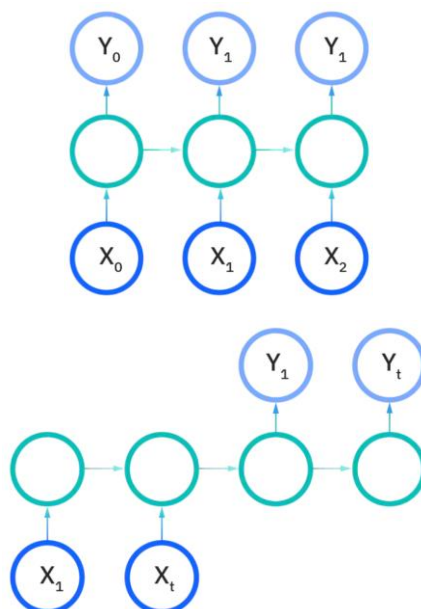


Рисунок 2.7 – Багато до багатьох

### Варіанти архітектури RNN.

**Двонаправлені рекурентні нейронні мережі (BRNN)** [43]: це варіант мережевої архітектури RNN. У той час як однонаправлені RNN можуть використовувати лише попередні входні дані для прогнозування поточного стану, двонаправлені RNN залучають майбутні дані для підвищення їх точності. Якщо повернутись до прикладу «відчуття погоди», модель зможе краще передбачити, що другим словом у цій фразі є «під», якщо вона знала, що останнє слово в послідовності — «погода».

**Довга короткочасна пам'ять (LSTM)** [44]: це популярна архітектура RNN, яка була представлена Зеппом Гохрейтером і Юргеном Шмідгубером як рішення проблеми зникнення градієнта. Тобто, якщо попередній стан, який впливає на поточний прогноз, не належить до недавнього минулого, модель RNN може бути не в змозі точно передбачити поточний стан. Як приклад, передбачення наступних слів, які виділені курсивом: «У Аліси алергія на горіхи. Вона не може їсти арахісове масло.» Контекст алергії на горіхи може допомогти нам передбачити, що їжа, яку не можна їсти, містить горіхи. Однак, якби цей контекст містився кількома реченнями раніше, RNN ускладнив би або навіть унеможливив зв'язок інформації. Щоб виправити це, LSTM мають «клітини» в прихованих шарах нейронної мережі, які мають три ворота: вхідний, вихідний і забутий. Ці ворота контролюють потік інформації, який необхідний для прогнозування виходу в мережі. Наприклад, якщо займенники статі, такі як «вона», повторюються кілька разів у попередніх реченнях, ви можете виключити це зі стану клітинки.

**Gated recurrent unit (GRU)** [45]: Цей варіант RNN схожий на LSTM, оскільки він також працює для вирішення проблеми короткочасної пам'яті моделей RNN. Замість використання інформації для регулювання «стану комірки» він використовує приховані стани, і замість трьох воріт він має два — шлюз скидання та шлюз оновлення. Подібно до шлюзів у LSTM, шлюзи скидання та оновлення контролюють, скільки та яку інформацію зберігати.

### **Microsoft Visual Studio.**

Microsoft Visual Studio [46] – це середовище розробки Microsoft, яке використовується для розробки різних типів програмного забезпечення, наприклад комп'ютерних програм, веб-сайтів, веб-програм, веб-служб і мобільних програм. Він містить інструменти завершення, компілятори та інші функції для полегшення процесу розробки програмного забезпечення.

Visual Studio IDE (інтегроване середовище розробки) – це програма, за допомогою якої розробники можуть писати та редагувати свій код. Його інтерфейс

користувача використовується для розробки програмного забезпечення для редагування, налагодження та створення коду.

### **Потужні інструменти програмування. IntelliSense, навігація за кодом та рефакторинг.**

Visual Studio прискорює розробку на мові JavaScript завдяки таким функціям, як автозавершення IntelliSense, відображення довідки по сигнатурах, міжпроектний рефакторинг і моментальна навігація за кодом (рис. 2.8).



Рисунок 2.8 – IntelliSense (компонент завершення коду)

### **Налагодження та діагностика. Швидкий пошук помилок у коді.**

Visual Studio забезпечує вбудовані висококласні можливості для налагодження коду JavaScript. Такі потужні функції, як зіставлення вихідного коду, дозволяють задавати точки зупинки безпосередньо в коді.

Профільовальники продуктивності значно спрощують пошук вузьких місць, пов'язаних із пам'яттю, у процесі виконання. Функція “Тільки мій код” дозволяє зосередитись на коді, який ви пишете зараз, та швидко виявляти проблеми, які уповільнюють роботу користувача, веб-дизайнера, конструктор класів і конструктор схем бази даних.

### **Вебінтерфейси.**

Можна розробляти вебзастосунки, використовуючи улюблені платформи. Visual Studio дозволяє інтегрувати популярні бібліотеки, такі як jQuery, Angular, Vue і React, з повнофункціональними вебплатформами, такими як ASP.NET Core,

щоб швидко створювати потужні програми для Інтернету. Можна об'єднати проекти інтерфейсів із серверними службами, щоб зручно переміщатися між проектами та розгорнути рішення повністю.

### **Node.js. Інструменти Node.js для Visual Studio.**

Можливе використання JavaScript для швидкої розробки вебслужб для багаторазового використання, які можна безпосередньо розгорнути в Azure (рис. 2.9). Доступні інтегровані висококласні інструменти для налагодження, профілювання та модульного тестування, а також широкий набір шаблонів проектів, які допоможуть швидко розпочати роботу.

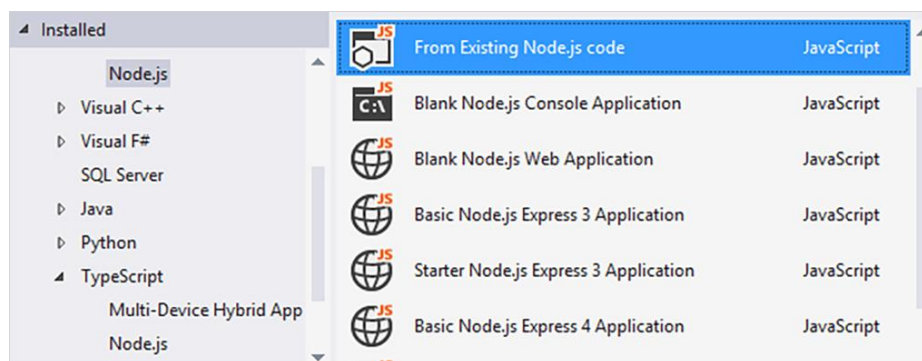


Рисунок 2.9 – Використання вебслужб

## **Висновки до розділу 2**

Проаналізувавши технології створення вебзастосунків, було обрано сценарій на стороні клієнта – це тип коду, який виконується або інтерпретується браузерами.

При цьому буде використано наступні мови програмування:

- HTML (мова розмітки гіпертексту);
- CSS (каскадні таблиці стилів);
- JavaScript.

Реалізація не міститиме бібліотек та фреймворків, що дозволить легко встановити вебзастосунок будь-якому користувачу, при цьому робота інтелектуальної системи буде максимально ефективною.



Для оптимальної роботи системи також буде проведено тестування наступних якостей:

- гарантія якості та тестування помилок;
- сумісність із кількома браузерами;
- безпека програми;
- продуктивність – тестування на навантаження та стрес.

Для процесу проектування інтелектуальної системи буде використано мову JavaScript, вона має багато переваг, для даного випадку особливими є:

- сучасний JavaScript компілюється в машинний код;
- сучасний JavaScript може використовувати апаратне прискорення.

Серед видів нейронних мереж буде використано рекурсивну нейронну мережу (RNN). У RNN результат обчислень на кожному етапі використовується як вихідні дані для наступного. Завдяки цьому вона може обробляти серії подій у часі або послідовності для отримання результату обчислень. Це особливо корисно, коли автомобілю потрібні дані про свій динамічний простір.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ТА ПОДОЛАННЯ ПЕРЕШКОД ТРАНСПОРТНИМИ ЗАСОБАМИ

#### 3.1 Механіка переміщення автомобіля

Першим етапом створення інтелектуальної системи була розробка механіки переміщення автомобіля (рис. 3.1). Вона включала в себе розрахунок тригонометричних формул для знаходження положення головного об'єкта у просторі та наділення його фізичними властивостями.

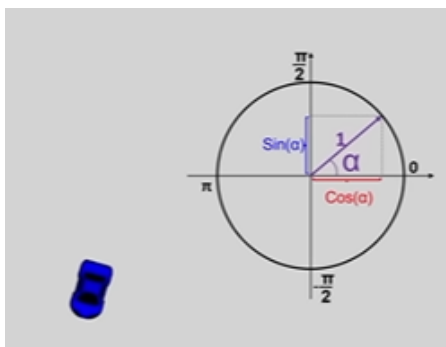


Рисунок 3.1 – Механіка переміщення автомобіля

Початкові точки  $x$  та  $y$  (рис. 3.2) знаходяться в центрі об'єкта, це реалізовано для визначення сторін автомобіля, таких як передня, задня, ліва та права його частини.

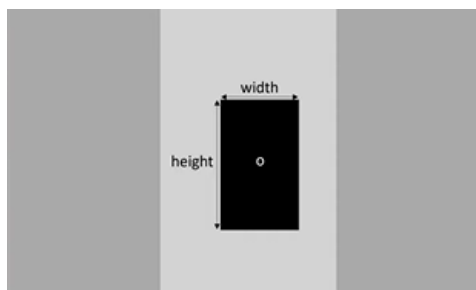


Рисунок 3.2 – Вигляд об'єкта автомобіля та його початкові точки  $x$  і  $y$

Для здійснення переміщення автомобіля (рис. 3.3) створено елементи керування. На ранньому етапі розробки керування здійснюється з використанням клавіш клавіатури. Для цього взято клавіші стрілок ( $\leftarrow$ ,  $\uparrow$ ,  $\downarrow$ ,  $\rightarrow$ ), а також створено слухача, який реагує на натискання клавіш та викликає з switch циклу потрібну дію. Для  $\leftarrow$  – поворот на ліво,  $\uparrow$  – рух вперед,  $\downarrow$  – зупинка, після зупинки рух назад,  $\rightarrow$  – поворот на право.



Рисунок 3.3 – Приклад переміщення авто

Переміщення вперед та назад виконуються за допомогою осі  $y$ . Вони представлені на рис. 3.4 та рис. 3.5.

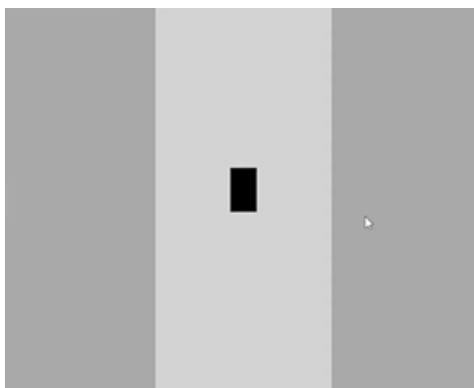


Рисунок 3.4 – Рух вперед

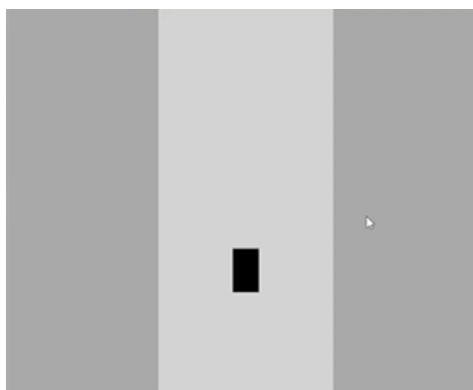


Рисунок 3.5 – Рух назад

Рух вперед, як і у звичайного реального авто, має максимальну швидкість, а також можливість змінювати поточну швидкість. Задня швидкість є значно меншою за звичайну. До головного об'єкта додано інерцію, прикладом даної фізичної властивості є гальмівний шлях під час зупинки автомобіля.

Зміна поточної смуги на іншу відбувається згідно з фізичними законами, тобто повинна відбуватись по діагональній траєкторії, тому застосовується співвідношення одиничного кола (рис. 3.6). Для поворотів на ліво та право при низькій швидкості або авто тільки починає рух, то у співвідношенні між осями  $x$  та  $y$ , більшим значенням виступає ось  $x$ , якщо ж авто перебуває на швидкості, то значення осі  $x$  не може бути більшим за  $y$ , тому що для переміщення в іншу смугу або поворот розраховуються таким чином, щоб авто безпечно виконало маневр без можливості перекинутись (рис. 3.6).

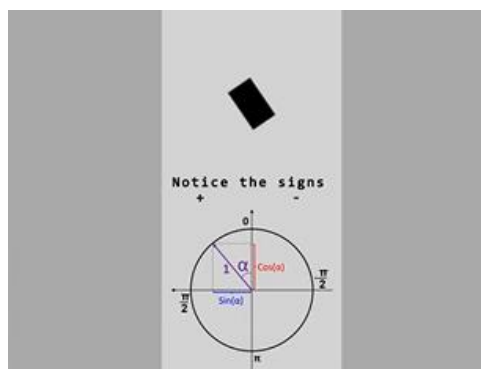


Рисунок 3.6 – Співвідношення одиничного кола

В результаті отримано коректне переміщення авто. Воно продемонстроване на рис. 3.7.

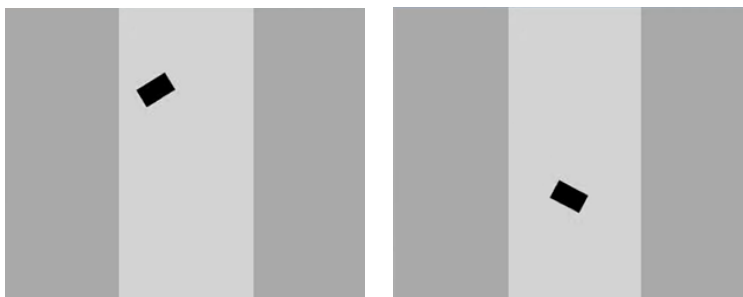


Рисунок 3.7 – Переміщення авто згідно співвідношення одиничного кола

### 3.2 Створення дороги та смуг

На даному етапі робочу площину розділено на 2 частини, перша, це візуалізація переміщення авто по дорозі та друга для візуалізації роботи нейронної мережі. Але на даний час розроблялась частина візуалізації переміщення авто, тому другу частину не видно на екрані через пустий блок.

У даній інтелектуальній системі дорога має дуже велику довжину, а не безкінечну, це зроблено для того, щоб зменшити частоту виникнення помилок у випадку з використанням безкінечного розміру (особливість мов програмування). Елемент дороги зображено на рис. 3.8.

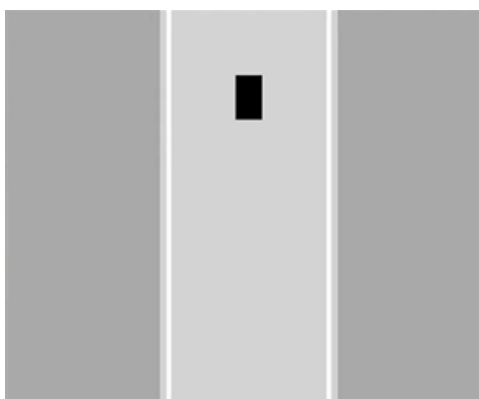


Рисунок 3.8 – Створена дорога

Наступним кроком був поділ дорожньої частини на 4 смуги (рис. 3.9). У мовах програмування рахунок починається з 0, тому першою роздільною смугою зображено смугу під значенням 0.

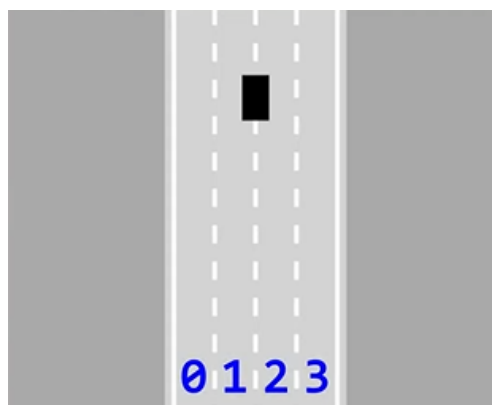


Рисунок 3.9 – Дорожня частина з роздільними смугами

Для кращої роботи системи та ефективнішого тестування кількість смуг зменшено до 3 та переміщено авто на центральну смугу (рис. 3.10).

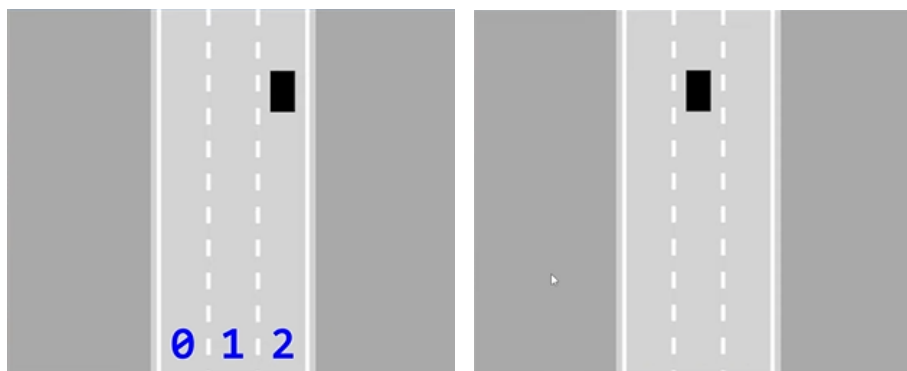


Рисунок 3.10 – Зменшення кількості смуг та розташування авто у центральній смузі

Одним серед важливих явищ було додавання крайніх меж дороги для уникнення зіткнень з відбійниками та виїзд авто на зустрічні смуги. Яскравими прикладами є автомагістралі та затяжний дорожній поворот (рис. 3.11).

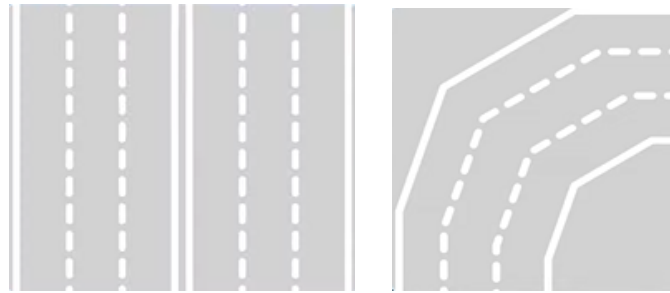


Рисунок 3.11 – Приклади автомагістралі та затаяжного дорожнього повороту

### 3.3 Штучні датчики

Даний етап є дуже важливим та складним через найважливішу частину, а саме процес збору інформації дорожньої обстановки. Для цього розроблялись штучні датчики, які мають сканувати зустрічні об'єкти чи перешкоди на своєму шляху та передавати дані в нейронну мережу. Побудовані вони таким же чином, як і поворотно-пересувний механізм авто (рис. 3.12), а саме з використанням співвідношення одиничного кола. Це зумовлено тим, що авто повинне мати дані про об'єкти чи перешкоди, які знаходяться поруч з автомобілем та мати можливість зреагувати в потрібний час та виконати потрібний маневр.

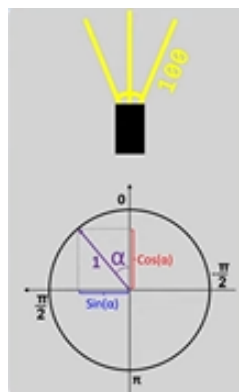


Рисунок 3.12 – Властивості датчиків

У датчиках довжина променів досягає 100 одиниць та може бути скорегована. Робота променів закладається у тому, що чим більше довжина променя, який

перетинає найближчий об'єкт чи перешкоду, тим більша чутливість у цього променю.

Датчики та їх промені зображені на рис. 3.13.

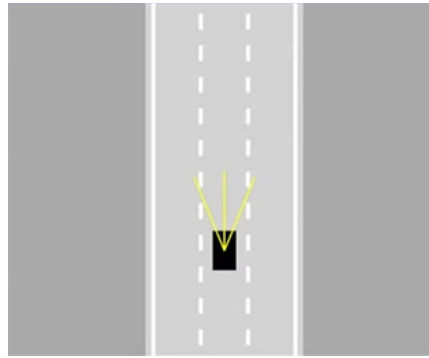


Рисунок 3.13 – Датчики та їх промені

У даній системі є можливість редагувати кількість датчиків та межу їх покриття. Використання даної можливості продемонстровано на рис. 3.14.

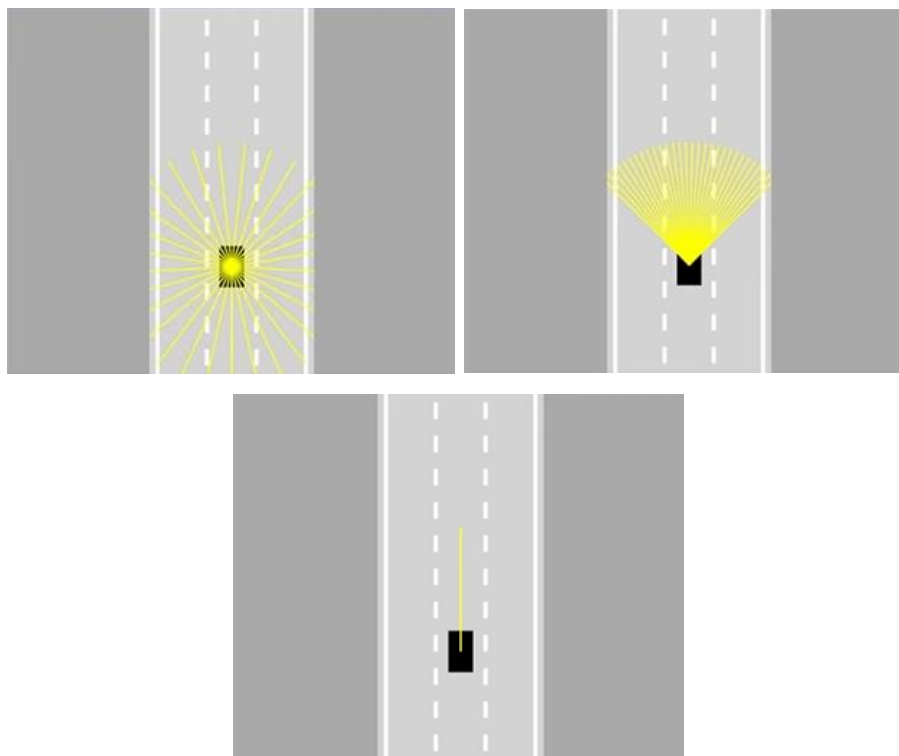


Рисунок 3.14 – Редагування кількості датчиків та межу їх покриття



Для отримання важливих та правильних даних, а також для оптимального навчання нейромережі обрано оптимальну кількість сенсорів – 5 одиниць (рис. 3.15). Ця кількість вважається оптимальною, тому що дані не перенасичують мережу, яка не має даних на початковому етапі та надають достатню кількість даних про навколишні об'єкти чи перешкоди для їх уникнення.

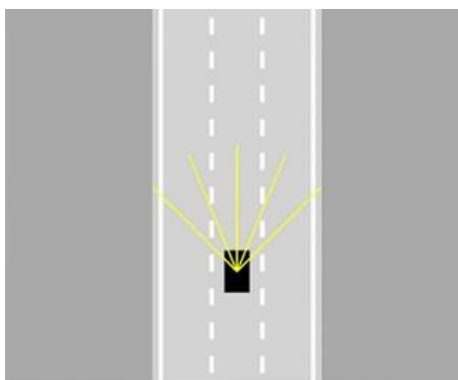


Рисунок 3.15 – Обрана оптимальна кількість датчиків

Наступним кроком потрібно було реалізувати реакцію датчиків на об'єкти та перешкоди, вони повинні їх зчитувати, як це показано на рис. 3.16.



Рисунок 3.16 – Зчитування датчиками об'єктів та перешкоди

Для цього взято розраховану довжину дотику променем до об'єкта чи перешкоди (рис. 3.17) та додано нову властивість датчикам, яка в свою чергу повинна надавати дані нейронній мережі та за її допомогою реагувати на них.

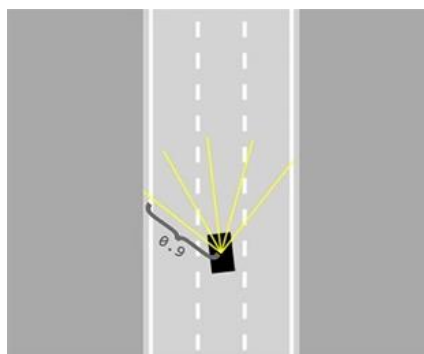


Рисунок 3.17 – Розрахунок довжини променю до перешкоди

Якісне (краще) відображення для візуального сприйняття перетину променів, які перетнули об'єкти чи перешкоди мають інший колір, чорний. Результат роботи датчиків показаний на рис. 3.18.

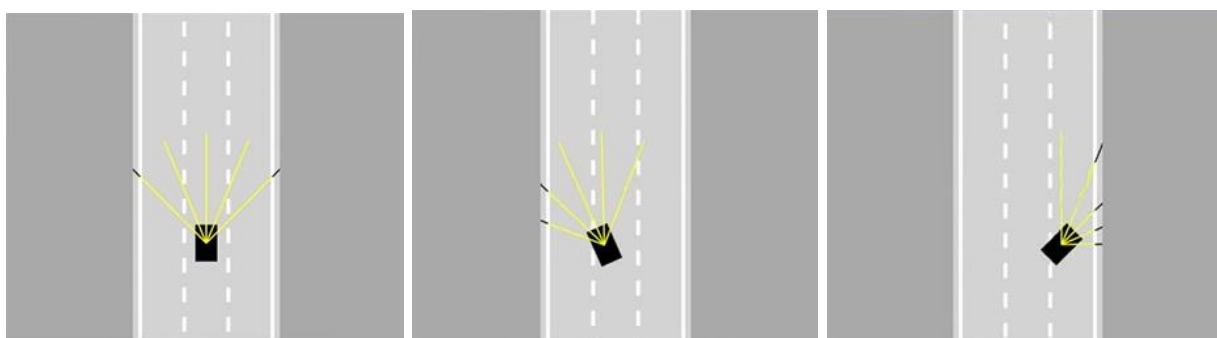


Рисунок 3.18 – Результат роботи датчиків

### 3.4 Виявлення зіткнень

Реалізацією даного завдання цільове (головне) авто перетворено в об'ємну фігуру. Для цього обраховано об'єм автомобіля за допомогою метода  $\text{hypot}()$ , це метод, який використовується для розрахунку об'єму фігур через розрахунок гіпотенузи або квадратного кореня з суми квадрата заданих двох сторін чи аргументів.

Використання даного методу представлено на рис. 3.19.

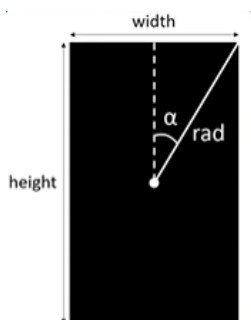


Рисунок 3.19 – Використання методу  $\text{hypot}()$

Обрахування потрібного кута об'єкта виконано за допомогою методу арккотангенсу (рис. 3.20). Він допомагає розрахувати кут маючи для цього довжину та ширину об'єкта.

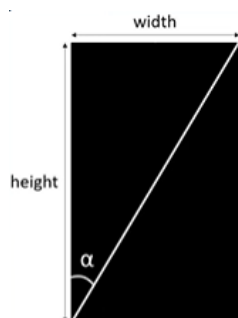


Рисунок 3.20 – Використання методу арккотангенсу

Наступним кроком було об'єднання методів розрахунку об'єму та положення авто (рис. 3.21).

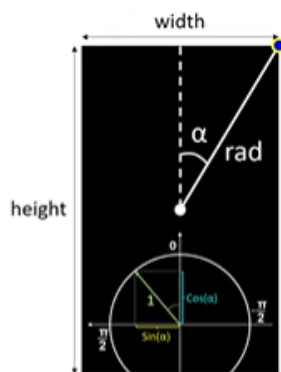


Рисунок 3.21 – Об'єднання розрахунків об'єму та положення авто

Після даних змін реалізовано реакцію на зіткнення з іншими об'єктами чи перешкодами. Працює це наступним чином, при зіткненні головний об'єкт фарбується в сірий колір, що позначає аварійну ситуацію. Дана реакція продемонстрована на рис. 3.23 та для порівняння представлена перед аварійна ситуація на рис. 3.22.

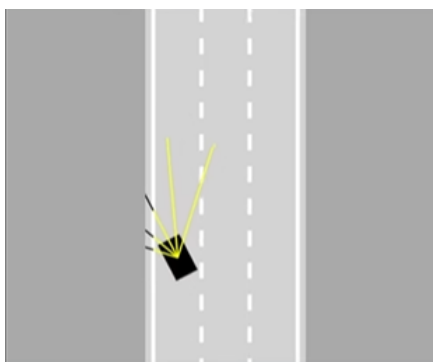


Рисунок 3.22 – Перед аварійна ситуація

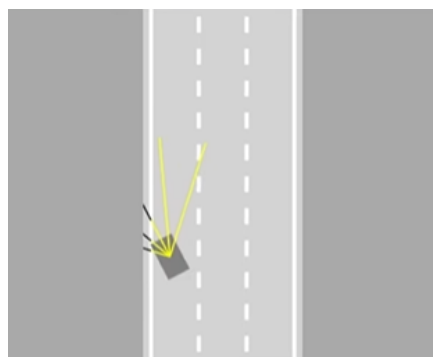


Рисунок 3.23 – Реакція об'єкта на аварійну ситуацію

### 3.5 Імітація трафіку

На даному етапі реалізовано додавання трафіку. Він імітує реальні умови на дорогах та призначений для навчання нейронної мережі долати перешкоди.

Для трафіку також додано властивість об'ємних фігур. Приклад роботи системи на цьому етапі показаний на рис. 3.24.

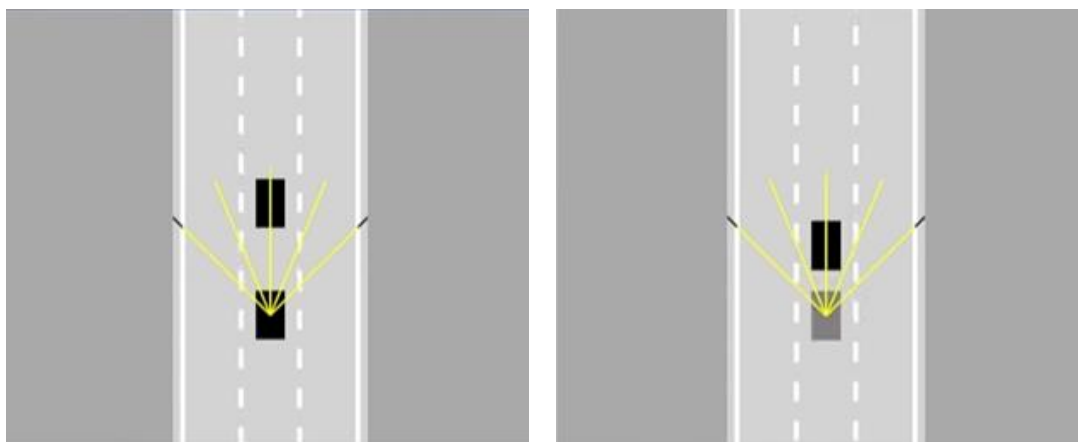


Рисунок 3.24 – Приклад роботи системи

З прикладів видно(3.5.1), що реакція на зіткнення авто працює, але промінь датчику не реагує на трафік. Тому наступним кроком було додавання логіки роботі датчикам та їх променям. Також для розрізнення цільового авто від трафіку, їм додано різні кольори, для головного синій колір, трафіку – червоний. Результат роботи показаний на рис. 3.25.

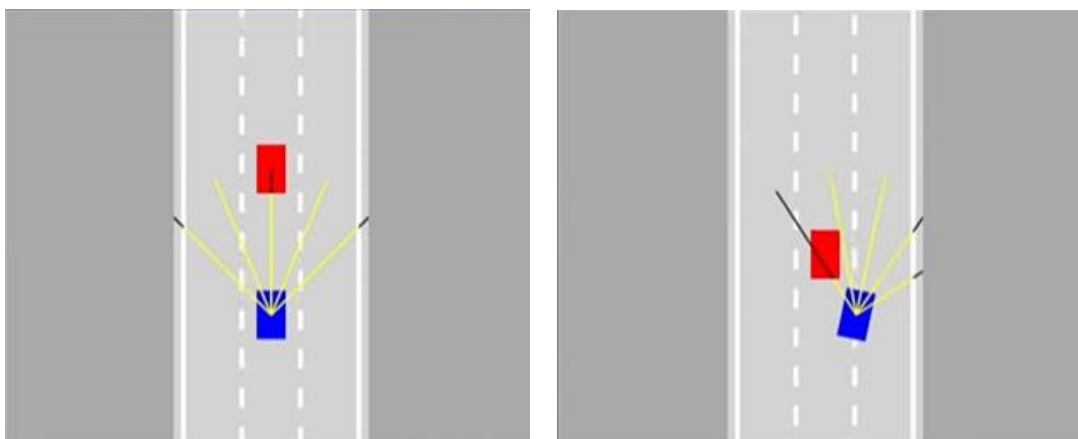


Рисунок 3.25 – Реагування датчиками трафіку

### 3.6 Нейронна мережа

Даний етап представляв собою створення нейронної мережі архітектурою у вигляді RNN.

В системі для ефективного поділу процесу виконання завдань нейронна мережа поділена на два рівні (рис. 3.26). Поділ реалізовано саме таким чином, тому що вхідний шар з сенсорами нульового рівня та вихідний шар з елементами керування першого рівня потрібно зв'язати з шаром для обчислення та прийняття рішень на основі вхідних даних, які після прийняття рішення повинні передаватись на шар з елементами керування.

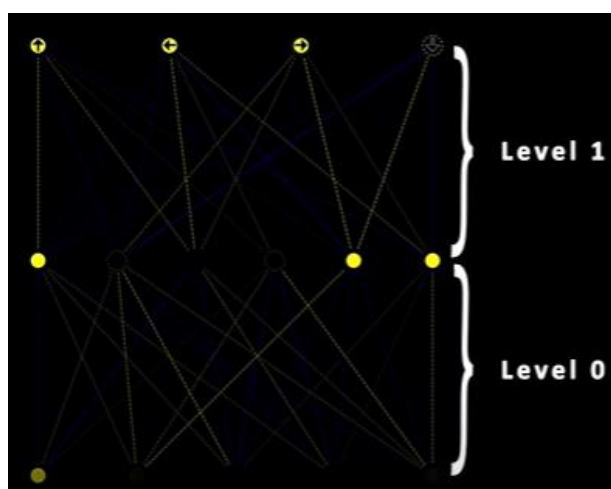


Рисунок 3.26 – Рівні нейронної мережі

У кожного рівня між нижнім та верхнім шарами утворюються нейромережеві з'єднання (рис. 3.27).

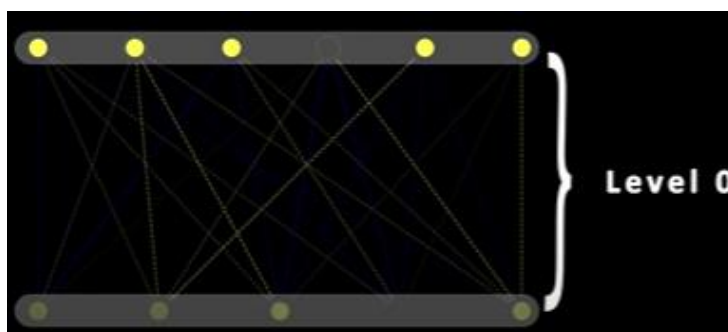


Рисунок 3.27 – Нейромережеві з'єднання нижнього та верхнього шарів рівня

Інтелектуальна система має 5 входів (сенсорних датчиків): «Лівий», «Діагональний лівий», «Передній», «Діагональний правий», «Правий» та 4 виходи, які можуть працювати як окремо, так і комбінуватись між собою: «Рух вперед», «Поворот вліво», «Поворот вправо», «Гальмування/рух назад» (рис. 3.28). Нейрони входів і виходів представляють собою масиви значень (рис. 3.29). Кожен вхідний нейрон з'єднаний з кожним вихідним нейроном рівня (рис. 3.30).

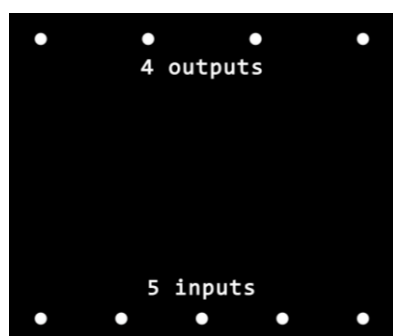


Рисунок 3.28 – Входи та виходи системи

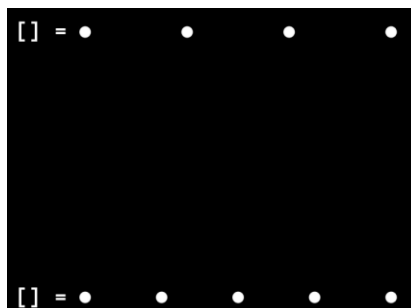


Рисунок 3.29 – Нейрони входів и виходів

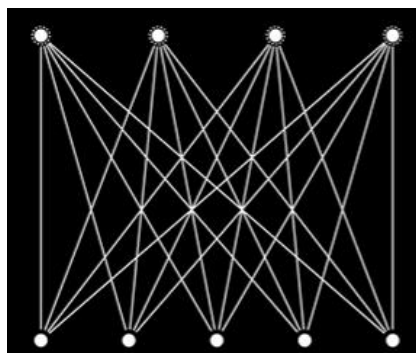


Рисунок 3.30 – З'єднання вхідних нейронів з вихідними

Як було раніше згадано, нейронна мережа має 3 шари, вхідний шар очікує вхідні дані від сенсорів та передає на прихований шар, а він в свою чергу приймає, обробляє та обчислює дані, після чого приймає рішення та передає його на вихідний шар, який підключений до елементів керування, виконує керування автомобілем (рис. 3.31).

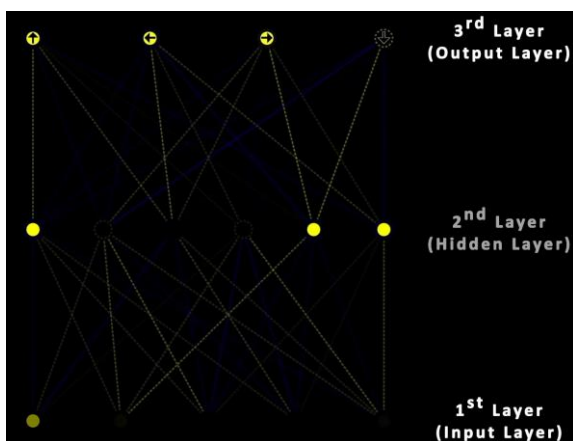


Рисунок 3.31 – Слої неромережевої системи

Для кожного вихідного нейрону усіх шарів застосовується формула (3.32), яка приймає дані та коефіцієнт зміщення.

$$W_0S_0 + W_1S_1 + W_2S_2 + W_3S_3 + W_4S_4 + b = 0, \quad (3.1)$$

де  $W_{0-4}$  – вагові значення сенсорів;

$S_{0-4}$  – вхідні дані з сенсорів;

$b$  – коефіцієнт зміщення.

Наступним кроком вихідний шар рівня обирає найбільше значення з розрахованого вхідного шару цього ж самого рівня за допомогою формули (3.1) та передає результат у наступний шар (рис. 3.32). Позитивні значення вважаються альтернативними рішеннями для вибору руху автомобіля, негативні, це значення які не можуть бути альтернативними рішеннями, тому що з високою вірогідністю призведуть до аварійних ситуацій.



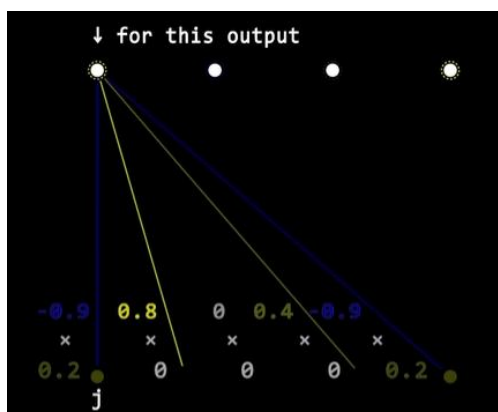


Рисунок 3.32 – Розрахунок та вибір вихідного слою

Останнім кроком даного підрозділу була візуалізація роботи нейронної мережі. Вона представлена на рис. 3.33.

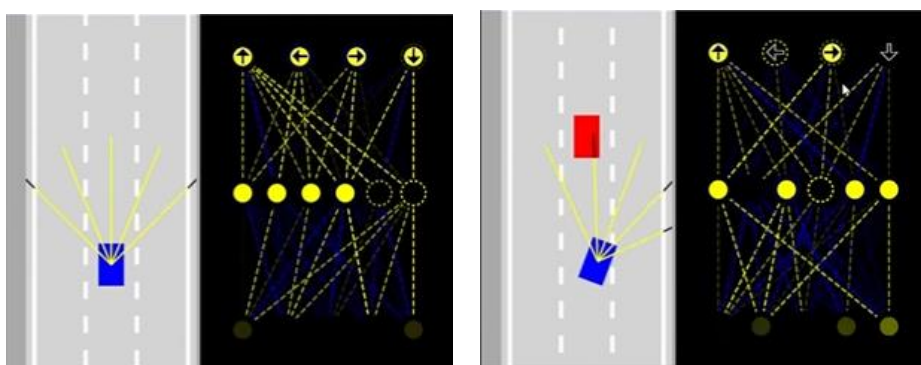


Рисунок 3.33 – Візуалізація роботи мережі

### 3.7 Розпаралелювання

На цьому етапі реалізовано можливість збільшувати та зменшувати кількість об'єктів для подолання перешкод. Тобто створюються паралельно головному об'єкту автомобіля, вказану кількість схожих, які мають такі ж властивості, але діють та приймають рішення незалежно один від одного. Це надає можливість використати всі можливі варіанти подолання перешкод на вирішення тої чи іншої ситуації на дорозі та додати знання в нейронну мережу для її навчання. Приклад роботи розпаралелювання показано на рис. 3.34.

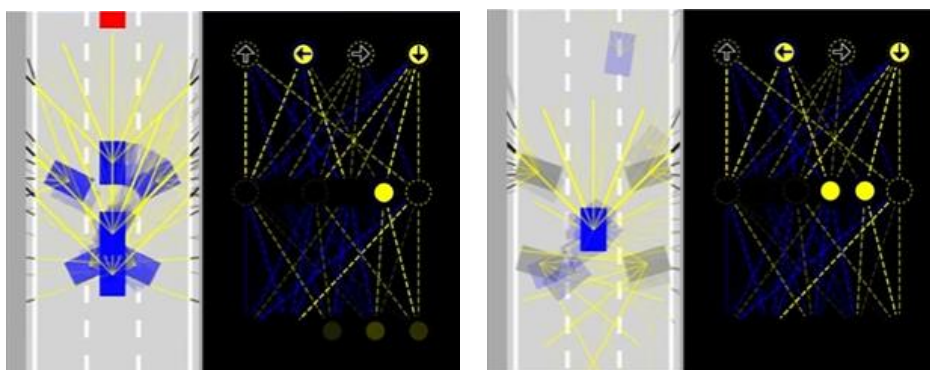


Рисунок 3.34 – Розпаралелювання об'єктів

Як видно з рис. 3.34, кількість сенсорів дуже велика, тому спостереження за об'єктами досить незручне, тому вирішено залишити відображення роботи датчиків цільовому автомобілю, яке приймає найкращі рішення щодо подолання перешкод. Таким чином система буде переходити між найкращими варіантами та надавати можливість спостерігати за ними. Після змін система отримала наступний вигляд, зображений на рис. 3.35.

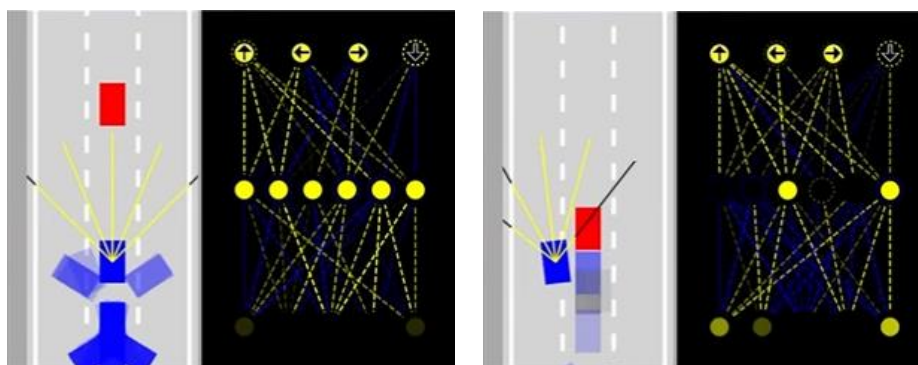


Рисунок 3.35 – Робота системи після виконаних змін

Для навчання нейронної мережі потрібні дані, тому для цього створено локальне сховище даних, на основі яких виконується навчання. Також створено можливість видалення даних, вона потрібна якщо збереглися помилкові дані для навчання.

Доданий функціонал системи показаний на рис. 3.36.

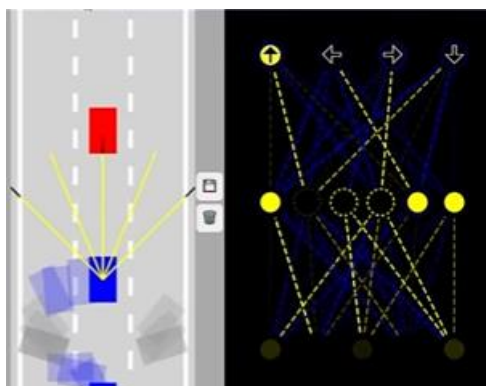


Рисунок 3.36 – Вигляд системи з новим функціоналом

### 3.8 Генетичний алгоритм

Генетичний алгоритм застосовується тільки для головних об'єктів або одного головного об'єкта, якщо виставлена кількість таких об'єктів рівна одному автомобілю. Працює даний алгоритм наступним чином, коли в локальному сховищі немає даних, то показники цільовим об'єктам надаються методом випадкових чисел. Далі виконується імітація уникнення перешкод та збереження даних найкращого набору дій у локальне сховище, після збереження даних, на їх основі навчається нейронна мережа та виконується процес мутації для кожного із головних автомобілів.

Процес мутації проходить наступним чином, для кожного головного автомобіля надаються показники випадковим чином, але у цьому випадку між отриманими даними та випадковими значеннями виконується операція добутку. Таким чином зберігаються дані про набір дій для уникнення попередньої перешкоди та з кінцевого розташування цільового об'єкта того ж рішення (найкращого), імітується уникнення наступної перешкоди. Такий процес виконується до тих пір, поки авто не проїде всі послідовні перешкоди.

У інтелектуальній системі параметри мутації можна редагувати. Це зроблено для ефективнішого зібрання даних та прийняття вірного рішення за найкоротший проміжок часу.

### 3.9 Результат роботи системи

На рис. 3.37 продемонстровано графічну роботу інтелектуальної системи. Вона включає в себе дві частини, які поділяють екран, перша показує рух авто, друга роботу нейронної мережі.

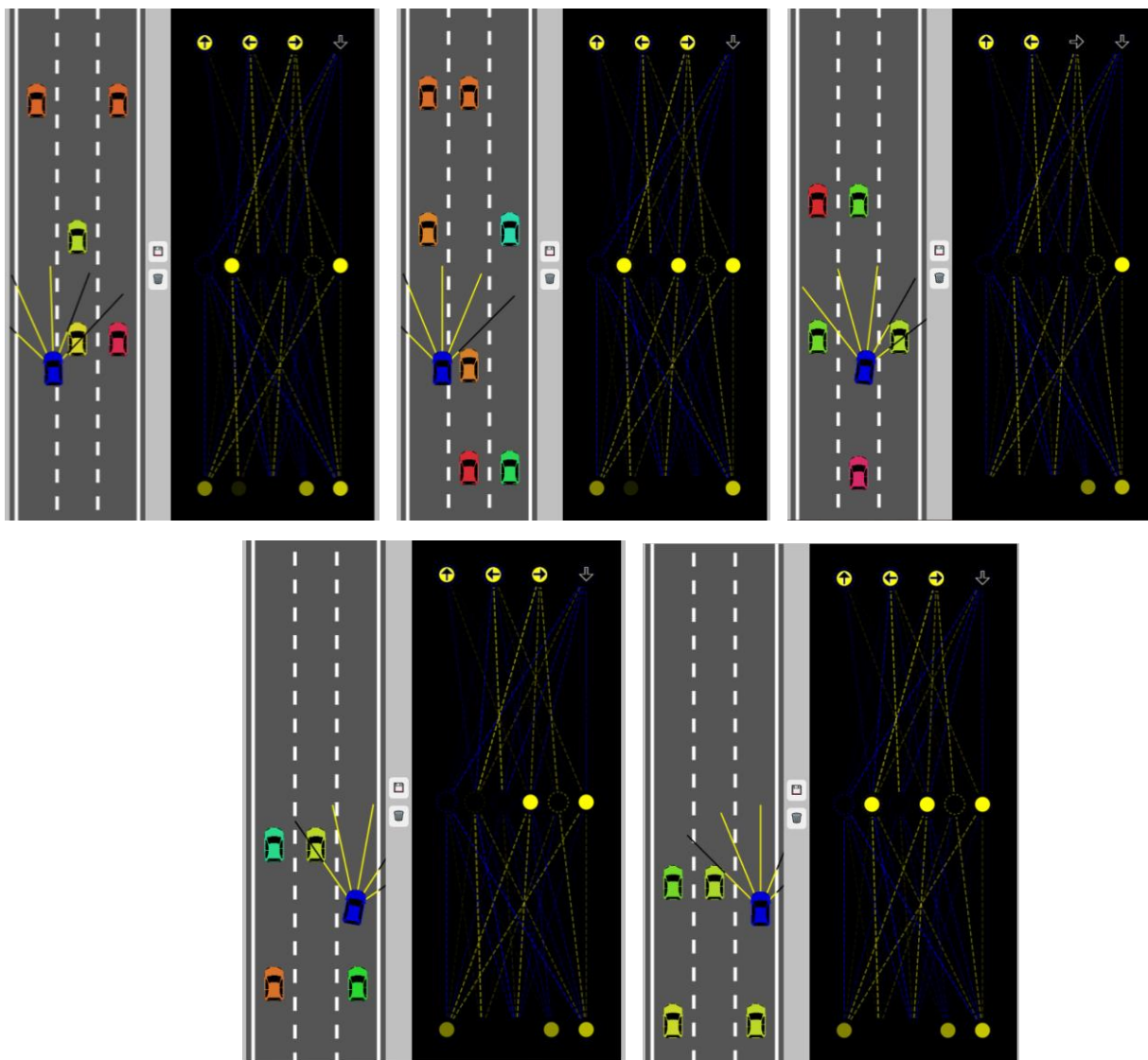


Рисунок 3.37 – Візуалізація роботи системи

### 3.10 Тестування роботи системи

Тестування інтелектуальної системи полягало у визначенні найкращих параметрів для уникнення перешкод автомобілем.

Змінювались наступні параметри:

- значення мутації в діапазоні від 0,1 до 0,3. Більше значення не використовувались, щоб не утворювалась сильна розбіжність між головними об'єктами, це досить сильно ускладнить процес навчання та подолання перешкод;
- кількість моделюючих головних об'єктів у діапазоні від 25 до 200 одиниць. Більшу кількість не використано через утворення підвисань, що в свою чергу надасть відповідну негативну реакцію на роботу інтелектуальної системи.

Результати тестування зображені в табл. 3.1.

Таблиця 3.1 – Результати тестування

Значення мутації	Кількість автомобілів для навчання	Час проходження рівня, хв.
0,1	200	25
	100	46
	50	70
	25	>120
0,2	200	10
	100	15
	50	22
	25	40
0,3	200	13
	100	21
	50	33
	25	56

З результатів тестування (табл. 3.1) можна зробити висновки, що найкращим показником мутації є 0,2 при кількості цільових об'єктів у 200 одиниць. Найгіршим себе показали параметри мутації 0,1 при кількості цільових об'єктів – 25 одиниць.

### Висновки до розділу 3

Під час виконання даного підрозділу було виконано наступні завдання:

- реалізовано механіку переміщення авто;
- створено дорогу та смуги;
- спроектовано штучні датчики;
- створено можливість виявлення зіткнень з перешкодами;
- реалізовано імітацію трафіку;
- спроектовано нейронну мережу;
- проведено розпаралелювання об'єктів;
- реалізовано генетичний алгоритм;
- продемонстровано роботу інтелектуальної системи;
- протестовано роботу системи.

Підрозділ механіки переміщення автомобіля включала в себе розрахунок тригонометричних формул для знаходження положення головного об'єкта у просторі та наділення його фізичними властивостями.

У підрозділі створення дороги та смуг, а саме створення дорожньої частини та розподіл її на смуги, після чого зроблено візуалізацію переміщення авто по смугам дороги.

Підрозділ штучних датчиків був дуже важливим та складним через найважливішу частину, а саме процес збору інформації дорожньої обстановки. Для цього розроблялись штучні датчики, які мають сканувати зустрічні об'єкти чи перешкоди на своєму шляху та передавати дані в нейронну мережу.

У підрозділі виявлення зіткнень представляв собою перетворення цільового (головного) авто в об'ємну фігуру та реалізацію реакції даного авто на зіткнення з іншими об'єктами чи перешкодами.

Підрозділ імітації трафіку був потрібен для додавання трафіку (перешкод) для їх уникнення.

У підрозділі нейронної мережі було створення нейронної мережі, а саме додавання 2 рівнів в мережу, кожен з яких мав нижній та верхній шари. Перший шар приймав дані з датчиків, другий приймав дані, обробляв їх, обчислював, приймав рішення та передавав дані на третій шар, котрий в свою чергу приймав їх та виконував керування автомобілем.

Підрозділ розпаралелювання був необхідний для створення вказаної кількості головних об'єктів. Це дало змогу змоделювати усі можливі варіанти уникнення перешкод.

У підрозділі генетичного алгоритму проводилась мутація головного об'єкта на основі отриманих даних.

Підрозділ демонстрації роботи інтелектуальної системи представляв собою графічну роботу системи, а саме уникнення перешкод та роботу нейронної мережі.

У підрозділі тестування роботи системи проводились тестування на оптимальність параметрів, а також проведено аналіз отриманих результатів и зроблено висновок.

## 4 ДІАГРАМИ ВИКОРИСТАННЯ ТА ПЕРСПЕКТИВИ ВДОСКОНАЛЕННЯ

### 4.1 Діаграми використання

#### Діаграма прецедентів.

Діаграма прецедентів (UML) – це основна форма вимог до системи/програмного забезпечення для нової недостатньо розробленої програми. Прецеденти визначають очікувану поведінку, а не точний метод реалізації. Коли прецеденти визначені, їх можна позначити як текстовим, і візуальним поданням (тобто діаграмою прецедентів). Ключовою концепцією моделювання прецедентів є те, що вона допомагає проектувати систему з погляду кінцевого користувача. Це ефективна техніка для повідомлення про поведінку системи з погляду користувача шляхом вказівки всієї поведінки системи, видимої ззовні.

Діаграма прецедентів зазвичай проста. Вона не показує деталі прецедентів:

- лише узагальнюють деякі відносини між прецедентами, дійовими особами та системами;
- показують порядок, у якому виконуються кроки задля досягнення цілей кожного прецедента.

Прецеденти є лише функціональними вимогами до системи. Інші вимоги, такі як бізнес-правила, вимоги до якості обслуговування та обмеження реалізації повинні бути представлені окремо, знову ж таки, з іншими діаграмами UML.

Діаграми прецедентів зазвичай розробляються на ранній стадії розробки, тому розробники часто застосовують моделювання варіантів використання для таких цілей:

- вказання контексту системи;
- фіксація вимог системи;
- перевірка системної архітектури;
- керування використанням та створення тестових прикладів;
- розробка аналітиками спільно з експертами у предметній галузі.



Основні елементи діаграми прецедентів:

- **суб'єкт (actor)** – будь-яка сутність, що взаємодіє з системою ззовні, або безліч логічно пов'язаних ролей, виконуваних при взаємодії з прецедентами;
- **прецеденти (use case)** – це опис множини послідовностей дій (включаючи їх варіанти), які виконуються системою для того, щоб актор отримав результат, який має для нього певне значення.

Між суб'єктами і прецедентами – основними компонентами діаграми прецедентів – можуть існувати різні відносини, які описують взаємодію екземплярів одних суб'єктів і прецедентів з екземплярами інших суб'єктів і прецедентів.

Нижче наведено діаграму прецедентів для інтелектуальної системи (рис. 4.1).

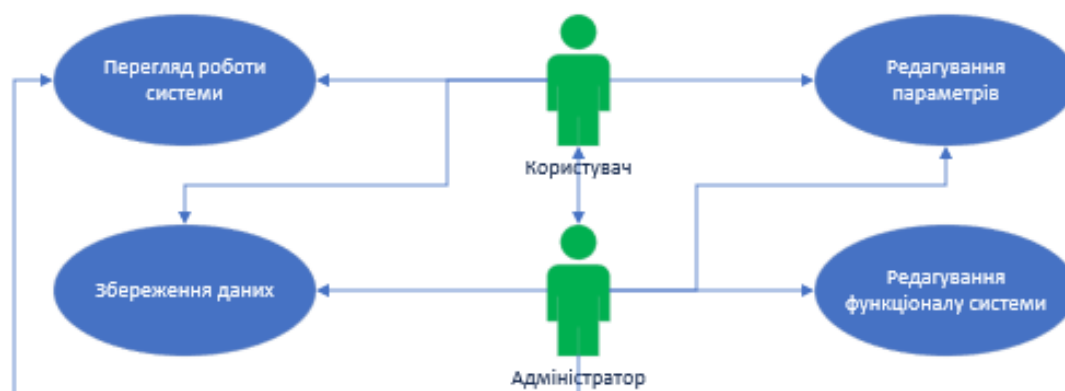


Рисунок 4.1 – Діаграма прецедентів для інтелектуальної системи

### Діаграма взаємодії.

Діаграми взаємодії (*Interaction Diagrams*) відображають взаємодію логічних елементів системи між собою в процесі виконання актором певного варіанту використання.

З терміну «Взаємодія» стає зрозуміло, що діаграма використовується для опису певного типу взаємодії між різними елементами в моделі. Ця взаємодія є частиною динамічної поведінки системи.

Ця інтерактивна поведінка представлена в UML двома діаграмами, відомими як діаграма послідовності та діаграма співпраці. Основне призначення обох діаграм подібне.

Діаграма послідовності наголошує на часовій послідовності повідомлень, а діаграма співпраці наголошує на структурній організації об'єктів, які надсилають і отримують повідомлення.

### **Призначення діаграм взаємодії.**

Метою діаграм взаємодії є візуалізація інтерактивної поведінки системи. Візуалізація взаємодії є складним завданням. Отже, рішення полягає у використанні різних типів моделей для охоплення різних аспектів взаємодії.

Діаграми послідовності та співпраці використовуються для відображення динамічної природи, але під іншим кутом.

Метою діаграми взаємодії є:

- фіксація динамічної поведінки системи;
- описання потоку повідомлень у системі;
- описання структурної організації об'єктів;
- описання взаємодії між об'єктами.

Діаграма послідовності містить чотири об'єкти (Клієнт, Замовлення, Спеціальне замовлення та Нормальне замовлення).

Друга діаграма взаємодії – це діаграма співпраці. Вона показує організацію об'єкта. На діаграмі взаємодії послідовність викликів методів позначається деякою технікою нумерації. Цифра вказує, як методи викликаються один за одним. Для опису діаграми співпраці взято ту саму систему керування замовленнями.

Виклики методів подібні до викликів діаграми послідовності. Однак різниця в тому, що діаграма послідовності не описує організацію об'єкта, тоді як діаграма співпраці показує організацію об'єкта.

Щоб вибрати між цими двома діаграмами, акцент робиться на тип вимоги. Якщо важлива часова послідовність, то використовується діаграма послідовності. Якщо потрібна організація, то використовується діаграма співпраці.

Діаграми взаємодії використовуються для опису динамічної природи системи. Щоб зрозуміти практичне застосування, потрібно зрозуміти основну природу послідовності та діаграми співпраці.

Основна мета обох діаграм подібна, оскільки вони використовуються для відображення динамічної поведінки системи. Однак конкретну мету важливіше прояснити і зрозуміти.

Діаграми послідовності використовуються для фіксації порядку повідомлень, що надходять від одного об'єкта до іншого. Діаграми співпраці використовуються для опису структурної організації об'єктів, які беруть участь у взаємодії. Однієї діаграми недостатньо для опису динамічного аспекту всієї системи, тому набір діаграм використовується для охоплення її в цілому.

Діаграми взаємодії використовуються, коли ми хочемо зрозуміти потік повідомлень і структурну організацію. Потік повідомлень означає послідовність потоку керування від одного об'єкта до іншого. Структурна організація означає візуальну організацію елементів у системі.

Можна використовувати діаграми взаємодії:

- моделювання потоку управління за часовою послідовністю;
- моделювання перебігу управління структурними організаціями;
- для передової техніки;
- для зворотного проектування.

Нижче наведено діаграму взаємодії для інтелектуальної системи (рис. 4.2).

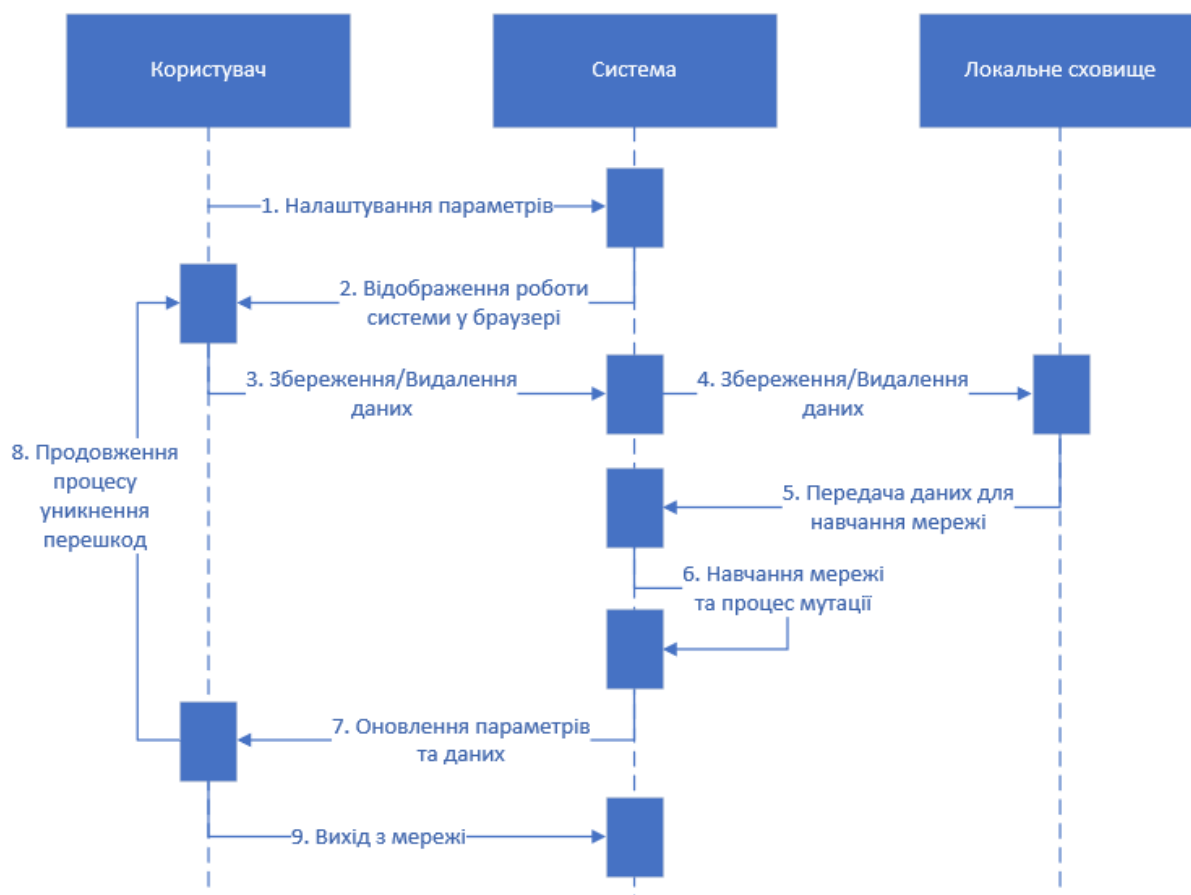


Рисунок 4.2 – Діаграма взаємодії системи

### Діаграма діяльності.

Діаграма діяльності (Activity diagram) відображає послідовність дій, що виконується в процесі реалізації певного варіанту використання або функціонування системи в цілому.

Діаграма діяльності є ще однією важливою діаграмою поведінки в діаграмі UML для опису динамічних аспектів системи. Діаграма діяльності – це, по суті, вдосконалена версія блок-схеми, яка моделює потік від однієї діяльності до іншої.

Діаграми діяльності описують, як діяльність координується для надання послуги, яка може бути на різних рівнях абстракції. Як правило, подія має бути досягнута деякими операціями, особливо якщо операція призначена для досягнення кількох різних речей, які вимагають координації, або як події в одному

варіанті використання співвідносяться одна з одною, зокрема випадки використання, де дії можуть збігатися і вимагати узгодження. Вона також підходить для моделювання того, як набір прецедентів координується для представлення бізнес-процесів:

- визначення потенційних прецедентів шляхом вивчення бізнес-процесів;
- визначення попередніх та післяумови (контексту) для прецедентів;
- моделювання робочих процесів між прецедентами та всередині них;
- моделювання складних робочих процесів в операціях над об'єктами;
- детальне моделювання складних дій на діаграмі діяльності високого рівня.

Діаграми діяльності вважаються аналогами блок-схем будь-якого алгоритму. Вони, як і діаграми станів та переходів, відображаються у вигляді орієнтованого графу, вершинами якого є дії, а ребрами – переходи між діями.

Іншими словами, якщо з вершини на діаграмі діяльності можна перейти до декількох інших вершин для всіх переходів необхідно визначити граничну умову.

Характеристика дії (action) для переходу не має сенсу, оскільки всі дії на цій діаграмі представлені вершинами графу.

Для діаграми діяльності характерними є наступні спеціальні стани:

- початковий стан – аналогічний до діаграми станів та переходів;
- кінцевий стан – аналогічний до діаграми станів та переходів;
- стан прийняття рішення – стан, в якому здійснюється прийняття рішення про перенаправлення потоку управління до одного зі станів, пов'язаних з даним станом.

Стан синхронізації – стан, в якому здійснюється розділення загального потоку управління на декілька гілок (чи навпаки, декілька гілок поєднуються в єдиний потік).

Основна діаграма діяльності зображена на рис. 4.3.

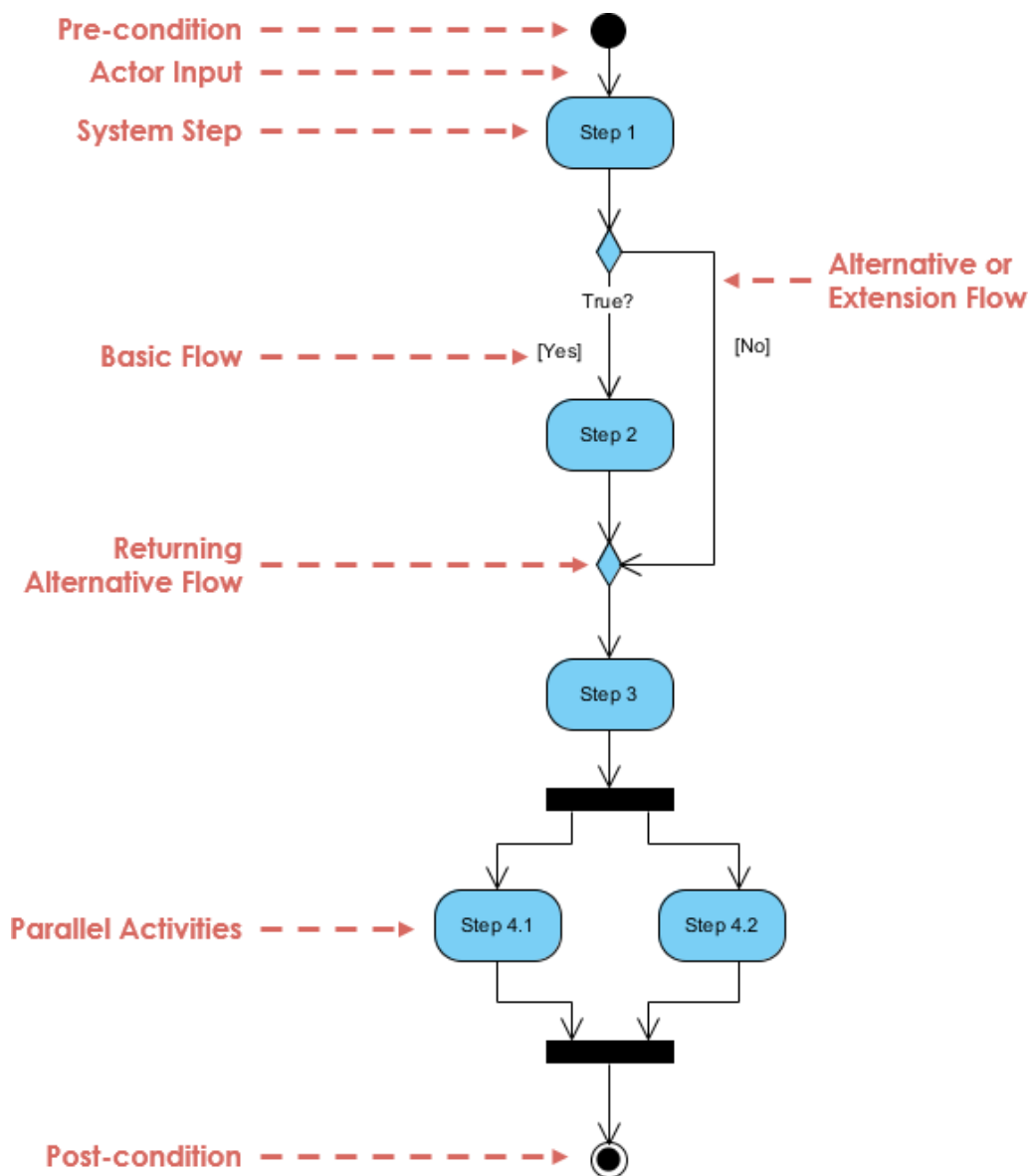


Рисунок 4.3 – Діаграма діяльності

Діаграма діяльності для інтелектуальної системи зображена на рис. 4.4.

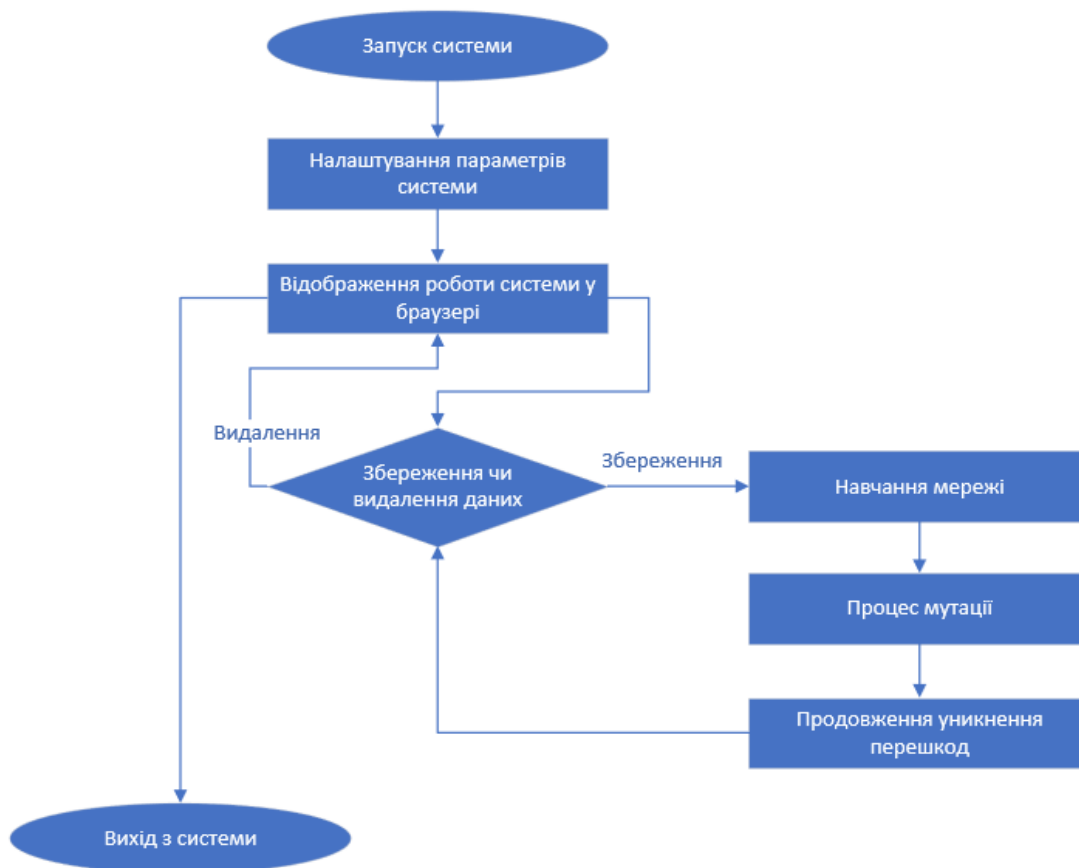


Рисунок 4.4 – Процес користування застосунком

## 4.2 Перспективи вдосконалення розробленої інтелектуальної системи

Для удосконалення даної інтелектуальної системи планується перевести відображення системи з 2Д виду в 3Д, також наділити об'єктів тривимірними властивостями. Дані зміни дозволять розробити функціонал для уникнення перешкод на рівні з автомобільними компаніями.

Приблизний вигляд системи планується та зображений на рис. 4.5:

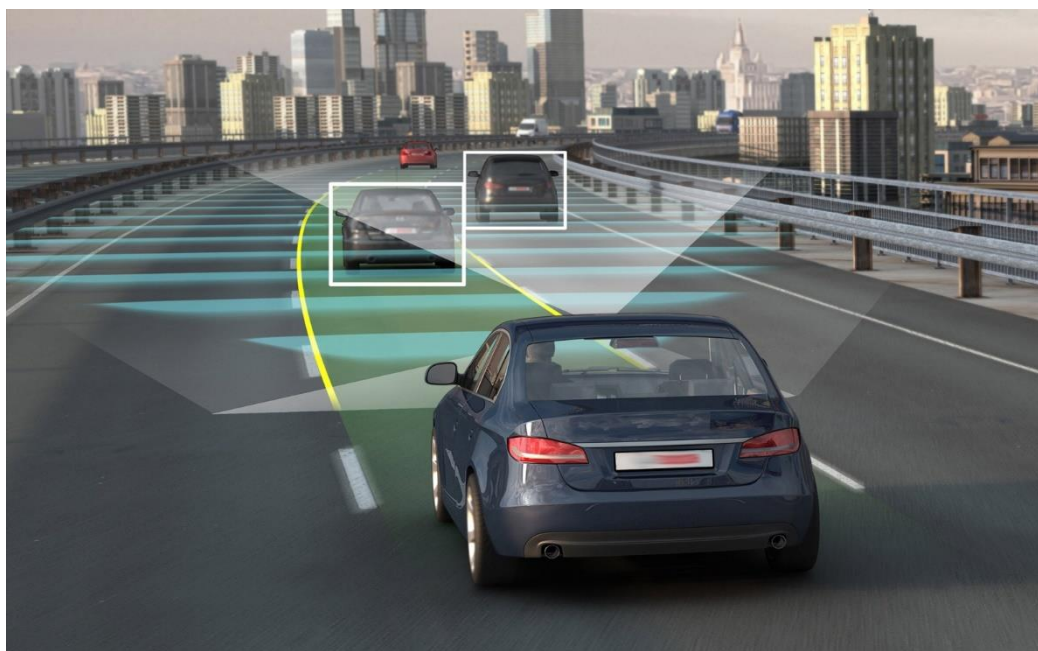


Рисунок 4.5 – 3Д вигляд системи

Наступним удосконаленням планується зміна транспорту, а саме додавання зустрічної смуги з зустрічним рухом. А також урізноманітнення видів перешкод.

Планується додати наступні:

- легкові транспортні засоби. Їх додано у даний список через те, що їх планується урізноманітнити марками-виробниками, моделями тощо;
- вантажні транспортні засоби. Даний вид транспортних засобів є одним із найнебезпечніших у наш час, тому нейронну мережу дуже необхідно навчити розпізнавати та оминати даний вид транспорту;
- регулюючі дорожні знаки. На даний час системи розпізнають не всі дорожні знаки, а деякі взагалі не розпізнають, тому для безпеки руху та можливості переходу на новий рівень автопілоту потрібно навчити нейронну мережу;
- вертикальну розмітку. Це більше стосується великогабаритних транспортних засобів для розуміння габаритів при входженні в тунелі, штучно розташовані об'єкти, котрі забороняють рух у даному напрямку;
- потяги, трамваї. Вважається найнебезпечнішим видом через мінімальну можливість вижити при зіткненні. Дуже важливо навчити нейронну мережу за



допомогою датчиків розпізнавати тип залізно-дорожнього переїзду (регулюємий чи нерегулюємий), сигнали світлофору та наближення чи відсутність поїзду/трамваю.

Цей етап дуже важливий для безпомилкового автопілотування автомобіля, то що в сучасних системах-аналогах великою проблемою є саме цей етап. Погане розуміння об'єктів заплутує автомобіль, а при помилковому розумінні, інтелект не знає, що йому робити в той чи інший момент часу, що й призводить до аварійних ситуацій;

Серед можливостей, котрі можна додати в систему можна виділити наступні:

- реагувати на показники автомобілів (світло фар, стопів, аварійної сигналізації і т.д.). Ця можливість потрібна для ефективного розуміння дорожньої ситуації в жвавому потоці, особливо це потрібно, коли попередній автомобіль різко вирішив зупинитись, для уникнення аварії в даній ситуації потрібно зчитувати світлові показники фар (у даному прикладі сигнал стоп фар);

- реагування на зустрічний транспорт та обгін, використовуючи смугу зустрічного транспорту. Планується додати до функціоналу не тільки односторонній рух, а й зустрічний. Для цього потрібно навчити систему розпізнавати зустрічні транспортні засоби, а при їх відсутності можливість виконувати обгін попередніх транспортних засобів;

- самостійний переїзд перетинів доріг. Як було сказано раніше, одним з найважчих елементів дороги для автопілоту є перетини. Вони можуть бути без розмітки, на них потрібно враховувати головну та другорядну смугу, показники світлофорів та/або регулювальників, а також напрямки руху зустрічних транспортних засобів;

- автоматичне переключення фар дальнього світла на ближнє для уникнення засліплення зустрічних автомобілів, а також навпаки при відсутності потреб для кращого освітлення дороги та зчитування інформації;

– додати використання GPS системи. Такі системи мають дорожню розмітку, тому для ефективності отримання інформації про поточне положення автомобіля потрібно порівнювати дані з датчиків та GPS систем. Також дуже потрібними є у зимовий період, коли дорожньої розмітки не видно;

– додати можливість автопарковки. Дуже популярним серед виробників автомобілів є саме ця можливість. Вона дуже допомагає людині у проблемах паркування самої людини. Бувають моменти, коли припаркуватись людині дуже важко, а інтелект маючи датчики міг би вирішити цю проблему і не пошкодити свій та сусідній автомобілі;

– під'їзд до власника за допомогою ключа автомобіля. Здебільшого розповсюджена у китайських автомобільних виробників. Найбільше використовується для виїзду з місця паркування. Зворотні проблеми людини паркування.

## ВИСНОВКИ

У наш час більшість автомобільних концернів сьогодні працюють зі автопілотом, який має характер дослідницьких прототипів.

Досліджено проблеми автопілоту, зокрема погане розпізнавання об'єктів за допомогою датчиків, їх неуніверсальність, та ринкову проблему.

Проведено аналіз систем аналогів, визначено їх обмеження та прийнято рішення для їх уникнення під час створення інтелектуальної системи. Серед систем аналогів розглянуто найбільші компанії, такі як Tesla Inc та Mercedes-Benz.

Для проектування інтелектуальної системи було виконано наступні задачі:

- проаналізовано сучасний стан задачі розпізнавання та подолання перешкод транспортними засобами;
- реалізовано обраний вид нейронної мережі, який дозволив збирати дані, передавати їх між собою, розраховувати результат та керувати автомобілем для уникнення перешкод;
- розроблено інтелектуальну систему для розпізнавання та подолання перешкод транспортними засобами у вигляді вебзастосунку;
- протестовано роботу розробленої системи, визначено ефективність та швидкість проходження змодельованих ситуацій шляхом зміни параметрів мутації та кількості об'єктів для розрахунку усіх можливих варіантів уникнення різноманітних перешкод.

У методичному розділі розроблено лабораторні роботи для курсу «Нейромереві методи обчислювального інтелекту» в галузі знань 12 «Інформаційні технології», зокрема 2 теми: метод LSTM для прогнозування продажів та класифікація зображень за допомогою нейронної мережі RNN.

У спеціальній частині з охорони праці було розглянуто питання охорони праці на робочому місці та цивільного захисту працівників ТОВ «Бріолайт-Україна». Зокрема розраховано комплексну бальну оцінку та описано заходи для забезпечення цивільного захисту на підприємстві.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Fazi B. M. Can a machine think (anything new)? *Autom Beyond Simul AI Soc* 34(4):813–824, 2019.
2. Burrell, J. How the machine “thinks”: Understanding opacity in machine learning algorithms. *Big Data & Society*, 2016.
3. Boer, A., van de Velde, R., & de Vries, M. *Self-driving vehicles (SDVs) & geo-information*, 2017.
4. Kowalczyk Z, Czubenko M. Intelligent decision-making system for autonomous robots. *Int J Appl Math Comput Sci* 21(4):671–684, 2011.
5. Czubenko M, Kowalczyk Z, Ordys A. Autonomous driver based on an intelligent system of decision-making. *Cognit Comput* 7(5):569–581, 2015.
6. Singh, S. Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey. Report No. DOT HS 812 115, 2015.
7. Beyerer J, Gheța M, Heizmann M, Belkin A. World modeling for autonomous systems. In: Dillmann R, Beyerer J, Hanebeck U, Schultz T (eds) *KI 2010. Advances in artificial intelligence*. Springer, Berlin, 2010, pp 176–183.
8. Bourguin P, Varela F. Towards a practice of autonomous systems. In: *Proceedings of the first European conference on artificial life*. MIT Press, Cambridge, 2018.
9. Matthaei R, Maurer M. Autonomous driving. A top-down-approach. *Automatisierungstechnik* 63(3):155–167, 2015.
10. Macrorie R, Marvin S, While A. Robotics and automation in the city: a research agenda. *Urban Geogr* 42(2):197–217. 2020.
11. Elfring J, Appeldoorn R, van den Dries S, Kwakkernaat M. Effective world modeling: multisensor data fusion methodology for automated driving. *Sensors* 16(10):1–28, 2016.
12. Stilgoe, J. Machine learning, social learning and the governance of self-driving cars. *Social Studies of Science*, 48(1), 2018, pp. 25–56.
13. Nyholm, S. The ethics of crashing with self-driving cars: A roadmap, II. *Philosophy Compass*, 13(17), e12506, 2018.

14. Shen, S. & Neyens, D. M. Assessing. Drivers' response during automated driver support system failures with non-driving tasks. *J. Saf. Res.* 61, 2017, pp. 149–155.
15. Alexey B, Chien-Yao W, Hong-Yuan L et al YOLOv4. Optimal speed and accuracy of object detection. *Computer vision and pattern recognition*, 2020.
16. Sato, T., Akamatsu, M., Shibata, T., Matsumoto, S., Hatakeyama, N., & Hayama, K. Predicting driver behavior using field experiment data and driving simulator experiment data: Assessing impact of elimination of stop regulation at railway crossings. *International Journal of Vehicular Technology*, 2013.
17. The 6 Levels of Vehicle Autonomy Explained URL: <https://www.synopsys.com/automotive/autonomous-driving-levels.html> (дата звернення 28.10.2022).
18. Autotech. 46 corporations working on autonomous vehicles. *CB Insights*, September 4, 2018.
19. Mercedes-Benz (Drive Pilot) URL: <https://group.mercedes-benz.com/innovation/case/autonomous/drive-pilot-2.html> (дата звернення 01.11.2022).
20. Tesla Inc (Autopilot) URL: <https://www.tesla.com/support/autopilot> (дата звернення 01.11.2022).
21. Google/Waymo (Waymo Driver) URL: <https://waymo.com/waymo-driver/> (дата звернення 01.11.2022).
22. Audi (Traffic Jam Pilot) URL: <https://audi.com/traffic-jam-pilot> (дата звернення 02.11.2022).
23. Baidu (Apollo) URL: <https://developer.apollo.auto/index.html> (дата звернення 02.11.2022).
24. General Motors (Super Cruise) URL: <https://www.gmc.com/connectivity-technology/super-cruise> (дата звернення 02.11.2022).
25. Toyota (Advanced Drive) URL: <https://toyota.com/advanced-drive> (дата звернення 03.11.2022).
26. Wepod (спільна розробка з NVIDIA) URL: <https://wepod.com> (дата звернення 03.11.2022).

27. Best Web Development Technologies to Use in 2022 URL:  
<https://www.clariontech.com/best-web-development-technologies-to-use-in-2022> (дата звернення 04.11.2022).
28. What are libraries in programming: URL: <https://www.idtech.com/what-are-libraries-in-coding> (дата звернення 04.11.2022).
29. The 5 best web application frameworks: How to choose URL:  
<https://techbeacon.com/app-dev-testing/five-best-web-application-frameworks-how-choose> (дата звернення 04.11.2022).
30. A Complete Guide To The 7 Phases of Web Development Life Cycle URL:  
<https://www.monocubed.com/web-development-life-cycle/> (дата звернення 04.11.2022).
31. Nicholas C. Zakas. Professional JavaScript for Web Developers 3rd Edition, 2022.
32. Kyle Simpson. You Don't Know JS : ES6, 2016.
33. WebGL: 2D and 3D graphics for the web URL:  
[https://developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API) (дата звернення 08.11.2022).
34. Brain.js: GPU accelerated Neural networks in JavaScriptfor Browsers and Node.js URL: <https://brain.js.org/#/> (дата звернення 08.11.2022).
35. TensorFlow.js URL: <https://www.tensorflow.org/js> (дата звернення 08.11.2022).
36. md5 URL: <https://www.npmjs.com/package/md5> (дата звернення 08.11.2022).
37. Guo Y, Liu Y, Oerlemans A, Lao S, Song W, Lew MS. Deep learning for visual understanding: a review. Neurocomputing 187:27–48, 2016.
38. Agostinelli F, Hoffman M, Sadowski P, Baldi P. Learning activation functions to improve deep neural networks. In: International conference on learning representations workshops, 2015.
39. Gerardus Blokdyk. Recurrent neural network: Real Life Actions, 2018.
40. Joish Bosco, Fateh Khan. Stock Market Prediction and Efficiency Analysis using Recurrent Neural Network, 2018.

41. Tarik Rashid. Recurrent Neural Network Model, 2013.
42. Convolutional neural network (CNN) URL:  
<https://www.techtarget.com/convolutional-neural-network> (дата звернення 13.11.2022).
43. Bidirectional Recurrent Neural Networks (BRNN) URL:  
<https://deeptai.org/machine-learning-glossary-and-terms/bidirectional-recurrent-neural-networks> (дата звернення 13.11.2022).
44. Long short-term memory (LSTM) URL: <https://intellipaat.com/lstm/> (дата звернення 13.11.2022).
45. Gated Recurrent Units (GRU) URL: <https://towardsdatascience.com/gru-networks> (дата звернення 13.11.2022).
46. JavaScript in Visual Studio Code URL:  
<https://code.visualstudio.com/docs/languages/javascript> (дата звернення 14.11.2022).
47. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин.
48. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу //Охорона праці. – 2001. –№ 12. – с. 12-20.
49. НПАОП 0.00-1.28-10. Правила охорони праці під час експлуатації електронно-обчислювальних машин.
50. Законі України «Про правові засади цивільного захисту». URL:  
<https://zakon.rada.gov.ua/laws/show/1859-15#Text> (дата звернення 27.01.2023)

## ДОДАТОК А

### Критерії бальної оцінки умов праці

Таблиця А.1

№ п/п	Фактор умов праці на робочому місці	Оцінка, бали					
		1	2	3	4	5	6
1	Температура повітря на робочому місці (РМ) у виробничому приміщенні, °С: - теплий період - холодний період	23...25 21...23	26...28 18...20	29...32 15...17	33...35 12...14	35...37 Нижче +12	>37 -
2	Відносна вологість повітря на РМ, %	40...50	55...60	61...75	76...85	Понад 85	-
3	Швидкість руху повітря на РМ, м/с	Менше 0,2	0,2...0,5	0,6...0,7	0,8...1,2	1,3...1,7	Понад 1,7
4	Освітленість на РМ, лк	≥ 300	240...300	160...230	100...150	60...90	30...50
5	Мінімальний розмір об'єкта розпізнавання, мм	> 1,0	1...0,3	< 0,3	0,005...0,3	< 0,05	-
6	Виробничий шум, перевищення ГДР, дБА	< 1	Рівно ГДР	1...5	6...10	> 10	> 10 з вібрацією
7	Інтенсивність теплового випромінювання, Вт/м <sup>2</sup>	≤ 140	141..1000	1001...1500	1501...2000	2001...2500	>2500
8	Токсична речовина, озон, кратність перевищення ГДК	-	≤ 1	1...2,5	2,6...4,0	4,1...6	> 6,0
9	Виробничий пил (паперовий), кратність перевищення ГДК	-	≤ 1	1...5	6...10	11...30	> 30
10	Робоче місце (РМ), поза та переміщення у просторі	РМ стаціонарне, поза вільна, маса переміщене вантажу ≤ 5 кг	РМ стаціонарне, поза вільна, маса переміщене вантажу > 5 кг	РМ стаціонарне, поза не вільна, до 25 % часу зміни у нахиленому положенні до 30°	РМ стаціонарне, поза вимушена – до 50 % робочої зміни	РМ стаціонарне, поза вимушена, незручна – більше 50 % робочої зміни	РМ стаціонарне, поза вимушена, незручна, нахили під кутом до 60 град більше 300 разів за робочу змін
11	Кількість важливих об'єктів спостереження	Менше 5	5...10	11...25	Понад 25	-	-



## Закінчення таблиці А.1

№ п\п	Фактор умов праці на робочому місці	Оцінка, бали					
		1	2	3	4	5	6
12	Тривалість зосередженого спостереження, % часу зміни	Менше 25	25...50	51...75	76...85	86...90	Понад 90
13	Тривалість повторюваних операцій, с	Понад 100	31...100	20...30	10...19	5...9	1...4
14	Змінність роботи	Ранкова зміна	Дві зміни	Три зміни	Нерегулярні зміни	-	-
15	Тривалість безперервної роботи за добу, годин	-	< 8	< 12	> 12	-	-
16	Режим праці та відпочинку	Обґрунтований, з включенням музики та гімнастики	Обґрунтований, без включення музики та гімнастики	Відсутність обґрунтованого режиму праці та відпочинку	-	-	-
17	Нервово-емоційне навантаження	Прості дії за індивідуальним планом	Прості дії за заданим планом з можливістю корегування	Складні дії за заданим планом з можливістю корегування	Складні дії за заданим планом при дефіциті часу	Відповідальність за безпеку людей	Індивідуальний ризик
18	Кількість рухів пальців на годину	< 360	360...720	721...1080	1081...3000	> 3000	-
19	Монотонність, тривалість операцій, які повторюються, с	> 100	31...100	20...30	10...19	5...9	1...4

## ДОДАТОК Б

### Залежність категорії умов праці від величини інтегральної бальної оцінки

Таблиця Б.1

<b>Діапазон інтегральної бальної оцінки</b>	<b>Категорія умов праці</b>	<b>Характер роботи</b>
До 18	I	Роботи, що виконуються в оптимальних умовах
19...33	II	Роботи, що виконуються в умовах, які відповідають гранично допустимим концентраціям (ГДК) і рівням (ГДР) санітарно-гігієнічних елементів, а також допустимим рівням психофізіологічних факторів
34...45	III	Роботи, що відхиляються від ГДК і ГДР та допустимих рівнів психофізіологічних факторів
45,7...53,9	VI	Робота у несприятливих умовах праці
54...59	V	Роботи, що виконуються в екстремальних умовах
Понад 59	VI	Роботи, що виконуються в екстремальних умовах