Міністерство освіти і науки України
Харківський національний університет
радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

«Дослідження алгоритмів та розробка системи автоматичного керування автомобілем»
(тема)

Виконала:
студентка 2 курсу, групи СШІм-19-1
Шаповалова К.Є.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник доц. Шергін В.Л.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«__» _____ 2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Шаповалової Катерини Євгенівни
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження алгоритмів та розробка системи автоматичного керування автомобілем»

затверджена наказом університету від 30 жовтня 2020р. № 1497ст

2. Термін подання студентом роботи до екзаменаційної комісії 16 грудня 2020р.

3. Вихідні дані до роботи система автоматичного керування автомобілем, відео результат роботи системи автоматичного керування автомобілем в симуляторі CARLA

4. Перелік питань, що потрібно опрацювати в роботі історія предметної галузі, методи та використовуванні технології для рішення основних питань, аналіз різних точок зору, тенденції при розробці, фактори, причини, котрі впливають на актуальність диплому, порівняльний аналіз, висновки про можливі шляхи вирішення для поставленого завдання, обґрунтування вибору напрямку дослідження, методи вирішення задачі і їх порівняльні оцінки, постановка задачі, реалізація поставленої задачі та результати досліджень

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) зображення розроблених автопілотних автомобілів, схеми: розташування сенсорів, системи автоматичного керування автомобілем, роботи супервайзера, модуля контролеру, схилу, модуля планування руху, модуля сприйняття навколишнього середовища, картографування навколишнього середовища; оригінальна карта з перешкодами, приклади з датасетів, передбачений трапецієподібний вхід дроселя протягом 20 секунд, обчислення траєкторії, пара стереозображень з CARLA, поєднані стереозображення, результати: тестування модулів системи, роботи карти глибини, роботи функції для вилучення особливостей, знайдених збігів, зміни траєкторії, отриманих координат дороги x, y, z, фільтрації, оцінки відстані до найближчих перешкод на дорозі, удосконаленої моделі виявлення смуг на дорозі, екстраполяції смуг, візуалізації: виявлення вихідних даних, пошуку шляху алгоритмом Дейкстри, алгоритмом пошуку A*, виявлення смуг на зображені; візуалізація: глибини зображення, результуючої теплової карти перехресної кореляції, обмежувального поля, траєкторії, набору вставок, передбачуваного простору для керування в 3D, поетапної побудови карти середовища зображення

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Теоретична частина	Доцент Шергін Вадим Леонідович		
Практична частина	Доцент Шергін Вадим Леонідович		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1	Отримання завдання на дипломну роботу	2.11.2020	Виконано
2	Аналіз предметної області і постановка завдання	8.11.2020	Виконано
3	Аналіз існуючих нейронних мереж	9.11.2020-10.11.2020	Виконано
4	Проектування нейроної мережі	11.11.2020-12.11.2020	Виконано
5	Реалізація системи автоматичного керування	13.11.2020-3.12.2020	Виконано
6	Обробка і оформлення результатів	4.12.2020	Виконано
7	Оформлення графічних матеріалів	5.12.2020	Виконано
8	Оформлення пояснювальної записки	6.12.2020-7.12.2020	Виконано
9	Попередній захист	8.12.2020	
10	Захист перед ЕК	16.12.2020	

Дата видачі завдання 2 листопада 2020р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 126 с., 54 рисунків, 2 таблиці, 1 додаток, 20 джерел.

CARLA, LIDAR, GPS/INS, СТАТИСТИКА, ГАУССОВСКИЙ РОЗПОДІЛ, НЕЙРОННІ МЕРЕЖІ, ЛІНІЙНА АЛГЕБРА, PYTHON

Об'єктом дослідження є автоматичне керування автомобілем. Для цього предметом дослідження була обрана система нейронних мереж завдяки якій проектується автоматичне керування автомобілем.

Метою роботи є розробка системи нейронних мереж для автоматичного управління автомобіля.

Методами дослідження є аналіз існуючих нейронних мереж, а також аналіз літератури та електронних ресурсів.

Результатом атестаційної роботи є нова система автоматичного керування автомобілем, на базі отриманих зображень та іншої прикладної інформації, такої як GPS / INS.

Дана робота має взаємозв'язок зі схожою американською системою від компанії Tesla.

Сферою застосування даної роботи є український автомобільний ринок.

Значимість роботи полягає в створенні української конкурентно здатної системі автоматичного управління автомобілем.

РЕФЕРАТ

Пояснительная записка: 126 с., 54 рисунков, 2 таблицы, 1 приложение, 20 источников.

CARLA, LIDAR, GPS/INS, СТАТИСТИКА, ГАУССОВСКОЕ РАСПРЕДЕЛЕНИЕ, НЕЙРОННЫЕ СЕТИ, ЛИНЕЙНАЯ АЛГЕБРА, PYTHON

Объектом исследования является автоматическое управление автомобилем. Для этого предметом исследования была выбрана система нейронных сетей благодаря которой проектируется автоматическое управление автомобилем.

Целью работы является разработка системы нейронных сетей для автоматического управления автомобиля.

Методами исследования является анализ существующих нейронных сетей, а также анализ литературы и электронных ресурсов.

Результатом аттестационной работы является новая система автоматического управления автомобилем, на базе полученных изображений и другой прикладной информации, такой как GPS/INS.

Данная работа имеет взаимосвязь с похожей американской системой от компании Tesla.

Сферой приминения данной работы является украинский авто рынок.

Значимость работы заключается в создании украинской конкурентно способной системе автоматического управления автомобилем.

ABSTRACT

Note explanatory: 126 p., 54 figures, 2 tables, 1 supplement, 20 sources.

CARLA, LIDAR, GPS/INS, STATISTICS, GAUSSIAN DISTRIBUTION, NEURAL NETWORKS, LINEAR ALGEBRA, PYTHON

The object of research is the automatic control of a car. For this, the subject of the study was a system of neural networks, thanks to which automatic control of a car is designed.

The aim of the work is to develop a neural network system for automatic vehicle control.

The research methods are the analysis of existing neural networks, as well as the analysis of literature and electronic resources.

The result of the certification work is a new automatic vehicle control system, based on the acquired images and other application information such as GPS / INS.

This work has a relationship with a similar American system from Tesla.

The scope of this work is the Ukrainian auto market.

The significance of the work lies in the creation of a Ukrainian competitively capable automatic vehicle control system.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АСК - Автоматична система керування;

ACC - Adaptive Cruise Control (Адаптивний круїз-контроль);

ADS - Automated Driving System (Автоматизована система водіння);

DDT - Dynamic Driving Task (Завдання динамічного водіння);

ES-EKF - Error-State Extended Kalman Filter (Розширений фільтр Калмана);

GPS - Global Positioning System (Система глобального позиціонування);

IMU - Inertial Measurement Unit (Інерційний блок вимірювання);

INS - Inertial Navigation Systems (Інерційні навігаційні системи);

LIDAR - Light Detection and Ranging (Виявлення та дальність світла);

ODD - Operational Design Domain (Домен операційного дизайну);

OEDR - Object and Event Detection and Response (Виявлення об'єктів, подій та відповідь на них).

ВСТУП

В даний час автомобільна промисловість робить гігантський перехід в майбутнє, в якому водій буде грати все меншу і меншу роль в управлінні своїм транспортним засобом поки у нього повністю не зникнуть всі обов'язки з управління.

В даному дипломі були досліджені різні рівні автоматизації водіння, передбачуваний вплив цих нових технологій на навколишнє середовище та безпеку дорожнього руху, була детально вивчена важливість нормативних актів і їх поточний стан, моральні аспекти впровадження цих технологій і можливі сценарії реалізації транспортних засобів з автоматичним управлінням. В ході дослідження було виявлено, що технології автоматичного керування транспортним засобом стикаються з безліччю проблем:

- автопілот повинен приймати рішення швидше в самих різних умовах, які також можуть включати в себе безліч моральних дилем;

- нейронні мережі мають значний потенціал в зниженні забруднення навколишнього середовища за рахунок оптимізації своїх маршрутів, стилів водіння шляхом взаємодії з іншими транспортними засобами, інфраструктурою і навколишнім середовищем;

- існує значний розрив між рівнем технологій автопілоту і діючими правилами; на щастя, в теперішній час спостерігається тенденція, де цей розрив постійно зменшується;

- у разі виникнення багатьох видів невідкладних ситуацій з управління нещасними випадками існує багато сумнівів щодо здатності прийняти правильних рішень.

З огляду на те, що дана галузь має надзвичайну швидкість розвитку і високу потрібність, дослідження є актуальним до крайнього терміну подачі. Технології автопілоту стають все більш витонченими і технічно доступними, а в деяких випадках їх можна використовувати і для

комерційних автомобілів. Відповідно до поточного етапу досліджень і розробок все ще неясно, як технології автопілоту зможуть справлятися з екстремальними і несподіваними подіями, включаючи їх моральні аспекти. Оскільки більшість дорожньо-транспортних пригод викликано людським фактором або бездіяльністю, очікується, що поява автопілоту скоротить кількість і тяжкість цих аварій, але в даний час існують не дуже багато доступних результатів випробувань, які здатні науково обґрунтувати це. Зростаюча тенденція до автоматизації транспортних засобів радикально змінить склад учасників автомобільної промисловості, оскільки мехатроніка стане не тільки додатковою частиною автомобільної промисловості, а й її невід'ємною частиною. Є вагомі підстави вважати, що автопілот буде працювати так само або краще в усіх відношеннях, ніж їх звичайні аналоги. Однак є підозра, що діючі правила не будуть встигати за розвитком технологій і тому іноді будуть перешкоджати розробки та тестуванню автопілоту.

Лідируюча компанія в даній сфері «Tesla» займається рішенням комерційної реалізації автопілоту та дослідженнями проблемних факторів в застосуванні цього продукту. Таким чином основні дослідження та комерційне застосування відбуваються в Америці. Тому додаткова ціль цього диплому - дослідження та розробка конкурентоздатного українського автопілоту, який в свою чергу може реалізуватися в комерційне українське підприємство з застосуванням найновіших технологій світу.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕНЬ

1.1 Історія предметної галузі

Автопілот пройшов довгу еволюцію, в ході якої він кардинально змінився з початкової ідеї до того, що ми звикли бачити сьогодні. В 1925 році Франсіс Юдіна продемонстрував автопілот під назвою «American wonder», фото даної події показано на рисунку 1.1.

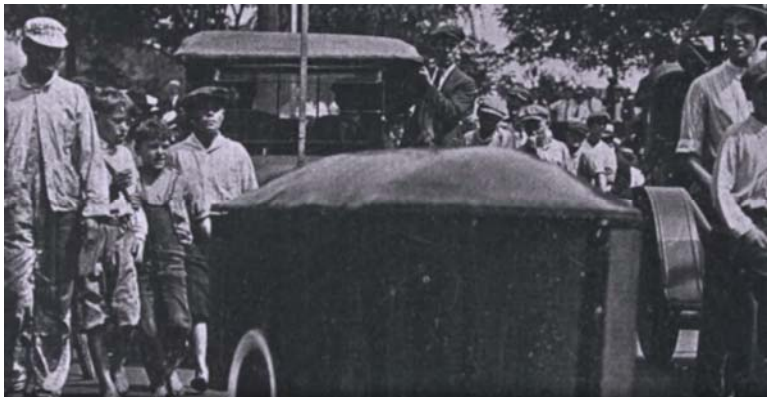

Рисунок 1.1 - Демонстрація автопілоту «American wonder», 1925 рік

Даний автомобіль їздив по вулицях Манхеттена лякаючи перехожих відсутністю водія за сидінням. В 1956 році, компанія General Motors створила рекламний відеоролик про машини з автопілотом, які плануються з'явитися на ринку до 1976 року. На той час компанія припускала, що водії автомобілів з автопілотом вже не будуть триматися за кермо автомобіля в ході їзди і будуть просто насолоджуватися безтурботною поїздкою, а пасажири можуть спілкуватися з людьми з інших машин за допомогою голосових функцій. Все це було досить оптимістично щодо того часу, але багато припущень з тих відео на сьогоднішній день стали доступними на ринку, такі як круїз контроль. Вимоги до АСК автомобілем йдуть з водіння в завантаженому трафіку. Одна з найбільш небезпечних речей в світі, яку

люди роблять на щоденній основі - керування автомобілем. Фактично, причиною більш ніж 94% [13] всіх інцидентів на дорозі сьогодні є людський фактор. Завдяки АСК автомобілем можна було б мінімізувати або повністю прибрати смерті в ході автокатастрофи, прибравши людський фактор із загальної картини на дорозі. Дана система мала б також багато й інших плюсів. Замість того, щоб концентруватися на дорожній смузі, поворотах та інших машинах, водій зможе відповідати на робочу пошту, їсти або відпочивати і зробити свій ранок набагато більш продуктивним. За допомогою даної системи, також можна поширити мобільність, яку дає водіння, на все людство, а не тільки на тих, хто здатен керувати автомобілем.

Наступні розробки були в Європі в 1980 році з ранніми моделями автомобілів, в основному частково автономними та працюючими на низьких швидкостях. У 1986 році Ернст Діккенс і його команда з Мюнхенського університету створили роботизований фургон, який міг рухатися повністю автономно без трафіку на дорозі, а в 1987 році він розігнався до швидкості 60 кілометрів на годину. На початку 1990-х його команда внесла значний вклад в проект Eureka Prometheus і розробила автономний Daimler Benz, використовуючи комп'ютерний зір, котрий фокусується на точках, які представляють інтерес в навколишньому середовищі. У них було більше 50 транспортерів, мікропроцесори того часу і ті ж імовірнісні підходи, які використовувалися у всій бортовий робототехніці, щоб зменшити кількість вхідних сигналів датчиків і при цьому реагувати на дорожні ситуації в реальному часі. Фото транспортера показано на рисунку 1.2 зліва, зображення камери з застосуванням нейронних мереж в ході поїздки показано на рисунку 1.2 справа.

Рисунок 1.2 - Фото транспортера під час тестування безпілотного руху зліва та зображення камери справа

В кінцевому підсумку вони проїхали 1600 км від Мюнхена до Копенгагена; середня відстань, яку транспорт міг проїхати без втручання людини, дорівнювала 9 кілометрам. У той же час в США університет Карнегі-Меллона був зайнятий створенням стабільної серії прототипів. У 1986 році з'явився перший автопілот, Navlab1, який розігнався по дорозі зі швидкістю 30 кілометрів на годину, використовуючи кілька сонячних робочих станцій. Фото Navlab1 показано на рисунку 1.3 зліва.

Рисунок 1.3 - Фото Navlab1 зліва та фото Navlab2 справа

Потім в 1990 році Navlab2 (показаний на рисунку 1.3 справа), модифікований Humvee, який міг автономно працювати як на бездоріжжі,

так і на дорозі, досяг автономної швидкості 110 кілометрів на годину на дорозі. Це підштовхнуло команду до чогось більш амбітного, і в 1996 році вони завершили своє турне по Америці без дотику до керма, яке мало 4800-кілометровий маршрут через континент, в ході якого був досягнутий вражаючий показник автономного водіння - 98,2% [15].

Незабаром після цього проект Каліфорнійського університету в Берклі став відомий завдяки успішним демонстрацій автономних взводів транспортних засобів, що працюють на виділених смугах руху для завантаженого транспорту на трасі I5. У 1992 році чотири автомобілі їхали в колоні на швидкісному шосе, покладаючись на магнітні маркери на проїжджій частині для точного відносного позиціонування, і продемонстрували, що використання такого конвою дозволяє заощадити на витратах на паливо і зменшити опір вітру. Успішні випробування тривали, і в 1997 році були завершені великомасштабні демонстрації від восьми до десяти автомобілів. В рамках проекту також було закріплено використання радіолокаційних систем і зв'язку між транспортними засобами в індустрії автоматизованих транспортних засобів, що дозволило отримати такі досягнення, як адаптивний круїз-контроль і екстрене гальмування.

Потім, в 2002 році, DARPA, Агентство перспективних оборонних дослідницьких проектів, оголосило про перше зі своїх завдань. Захід було проведено в 2004 році, і DARPA запропонувало переможцям приз в 1 мільйон доларів, якщо вони зможуть побудувати автопілот, який зможе проїхати 142 миль через пустелю Мохаве [1]. Хоча на першому етапі тільки кілька команд зійшли з лінії старту, а команда Карнегі-Меллона, яка зайняла 1 місце, не доїхала 7 миль до кінця, було ясно, що завдання проїхати 140 миль по пустелі без будь-якої допомоги людини дійсно було можливо виконати. Фото машини одного з учасників даного змагання показано на рисунку 1.4.

Рисунок 1.4 - Учасник першого змагання автомобілів з автопілотом в пустелі від DARPA

У другому випробуванні DARPA в наступному році 5 з 23 команд успішно пройшли трасу без будь-якого втручання людини. Стенфордський автомобіль виграв випробування, за ним був Carnegie Mellon Sandstorm. Пізніше, в рамках програми 2007 року, названої DARPA Urban Challenge, університети пропонували продемонструвати свої автопілоти на завантажених дорогах з професійними каскадерами. На цей раз, після 30-хвилинної затримки, коли екран jumbotron заблокував їх транспортний засіб від прийому сигналів GPS, команда Carnegie Mellon вийшла на перше місце, а автомобіль Stanford Junior посів друге місце.

Дані завдання від DARPA змінили уявлення громадськості і технічної та автомобільної промисловості про можливість повної автоматизації транспортних засобів. Після цього починає відкриватися новий ринок АСК автомобілем. Google запрошує керівників команд з Карнегі-Меллона і Стенфорда, Кріса Томпсона і Майка Монте-Карло, щоб вони просували свої проекти на дорогах загального користування. До 2010 року автомобіль Google проїхав понад 140 тисяч миль в Каліфорнії, а пізніше написав у блозі, що вони впевнені в скороченні вдвічі кількості смертей в результаті дорожньо-транспортних пригод [19]. Їх прагнення до

цієї мети триває і донині: станом на жовтень 2020 роки команда подолала понад 10 мільйонів миль. Фото автомобілів з автопілотом від компанії Google показано на рисунку 1.5.

Рисунок 1.5 - Фото автомобілів з автопілотом від компанії Google

Приблизно в той же час, коли відбулося перше автономне тестування в США, два помаранчевих фургона від компанії VisLab завершили міжнародну автономну задачу, проїхавши весь маршрут від університету Парми до Шанхаю, який склав близько 15000 кілометрів з мінімальним втручанням водія [1]. Фото оранжевого фургона показано на рисунку 1.6.

Рисунок 1.6 - Фото оранжевого фургона, США, перше автономне тестування

Рік по тому Volvo продемонструвала концепцію автопоїзда, в якій один транспортний засіб міг керувати кількома іншими транспортними засобами, що перебувають позаду нього, тим самим зменшуючи затори на дорогах і підвищуючи комфорт водія. Водії легкових автомобілів могли вибрати чільну вантажівку, щоб слідувати за нею, але близьке проходження призводило до частих пошкоджень транспортного засобу в умовах поганої видимості і в дощову погоду. Фото демонстрації концепції автопоїзда від компанії Volvo показано на рисунку 1.7.

Рисунок 1.7 - Фото демонстрації концепції автопоїзда від компанії Volvo

Завдяки прискоренню темпів розробки все більше і більше компаній наполягали на створенні законодавчої бази для дорожніх випробувань. У 2012 році департамент автотранспортних засобів Невади видав Google першу в історії ліцензію на випробування автономних транспортних засобів, що означало, що тепер вони можуть автономно тестувати автомобілі на дорогах загального користування. Каліфорнія незабаром пішла слідом цьому прикладу і виставила обов'язковою вимогою публікувати різні статистичні дані про проведені тестах і показниках втручання людини. У 2013 році, Mercedes Benz продемонстрував свої можливості автономного водіння, пройшовши той же 106-кілометровий маршрут в автономному режимі [20]. Завдяки тому, що кілька гравців

продемонстрували життєздатні дорожні демонстрації, інтерес громадськості до автономного водіння різко зріс. Виставка побутової електроніки в Неваді стала щорічною демонстрацією найбільших технологій автономного водіння і найсміливіших модифікацій автомобілів.

Приблизно через рік Google продемонстрував свій автомобіль-світлячок (рисунок 1.8), транспортний засіб, розроблений з нуля для автономної роботи зі швидкістю до 40 кілометрів на годину [19]. Без керма і педалей пасажири можуть замість цього сісти і розслабитися, і натиснути лише одну кнопку зупинки, якщо вони відчувають себе незручно. Тим самим Google показав нам, як може виглядати майбутнє автономного водіння.

Рисунок 1.8 - Автомобіль-світлячок від компанії Google

Компанія Tesla тим часом не відставала. Приблизно наприкінці 2015 року ця фірма представили свій набір автономних функцій в програмі під назвою «Autopilot». До 2016 року вони продемонстрували, як автопілот може самостійно паркуватися і може бути викликаний за допомогою кнопки. Це була найбільша демонстрація частково автономних функцій на той час і Tesla швидко подолала мільйони кілометрів, проїхавши на своєму автопілоті. Фото роботи безпілотника від компанії Tesla показано на рисунку 1.9.

Рисунок 1.9 - Фото роботи безпілотника від компанії Tesla

Прогрес не зменшував обертів. В цей самий час ставало популярним поняття автономних парків таксі. Стартапи, такі як Zoox і nuTonomy, почали лідирувати. Фактично, таксі фірми nuTonomy (рисунок 1.10) вперше почали працювати в Сінгапурі в 2016 році.

Рисунок 1.10 - Фото таксі nuTonomy в Сингапуре

Час між втручанням людини продовжує зростати, оскільки самі передові системи долають тисячі кілометрів без будь-якого втручання. Продуктивність на людському рівні, схоже, не за горами. Незважаючи на це, щоб домогтися трансформаційних змін, які, як очікується, принесуть в суспільство безпілотні автомобілі, при розробці АСК автомобілем потрібно вирішити основні проблеми створення такої системи.

1.2 Основні проблеми створення АСК автомобілем

Основні проблеми створення АСК автомобілем можливо поділити на:

- дороге фінансування досліджень;
- високі вимоги до якості роботи АСК автомобілем;
- конфіденційність результатів досліджень компаній;
- неточність та не універсальність моделей сприйняття.

Для того щоб проводити дослідження в реальному середовищі потрібно дороге обладнання, таке як: всі можливі датчики (Lidar, Sonar, камери і так далі), змінений автомобіль. Змінений автомобіль повинен додатково до системи керування водієм людиною мати необхідне обладнання для АСК автомобілем, тобто система повинна мати доступ до управління швидкістю автомобіля, поворотів і так далі, мати додаткову систему охолодження для обладнання контролю системою, блок генератора для системи, а також обладнання, в разі потреби, для передачі управління від АСК автомобілем до водія людини.

Навіть маючи дане обладнання, в витрати також потрібно враховувати зміну поломаного або неправильно робочого обладнання, що трапляється коли: погані умови навколишнього середовища, помилка в АСК автомобілем, що привела до пошкодження обладнання, наприклад, зіткнення зі стіною, недораховані потреби до обладнання, такі як, збільшення радіусу Sonar для більш акуратного паркування, непередбачувані системою події: перехід дороги тваринами, котра система не вміє впізнавати і таким чином не може прийняти правильне рішення.

Оскільки від якості роботи АСК автомобілем залежить життя людини, котра перебуває в автомобілі або знаходиться поруч, до таких систем існують високі вимоги до якості роботи АСК автомобілем. Перша аварія зі смертельним результатом за участю Tesla Model S сталася у Флориді в 2016 році. Це було викликано одночасною відмовою датчиків

камери і радара правильно ідентифікувати вантажівку, що повертає ліворуч, і неуважний водій, що звик до машини, яка сама про себе дбає. Ця невдача продемонструвала небезпеку і неможливість повністю покладатися на моніторинг автономних систем з боку людини. А в 2018 році, відбулася сумнозвісна аварія Uber, яка забрала життя пішоходів в Арізоні (рисунок 1.11). Цей інцидент викликав шок серед спільноти тестування автономних транспортних засобів, і Uber припинив тестування до тих пір, доки не будуть виявлені та зрозумілі всі подробиці події в результаті незалежного розслідування, проведеного державними органами безпеки дорожнього руху.

Рисунок 1.11 - Новини про ДТП за участю безпілотного автомобіля компанії Uber

Дані випадки та ризик збільшення аварій на дорозі спонукали американську владу розробити стандарт NHTSA [2], лише відповідаючи котрому можливо отримати дозвіл на водіння АСК автомобілем по громадській дорозі. Для отримання даної ліцензії система автопілоту повинна доказати свою працездатність та безпеку, а доки навчання системи може проходити лише на приватній території, котра в свою чергу повинна бути досить великою та мати способи надати різні умови

навколишнього середовища, щоб бути достатньо наближеною до умов на громадській дорозі.

Оскільки успіх та швидкість реалізації АСК автомобілем впливає на прибуток компаній, котрі цим займаються, в даній предметній галузі конфіденційність результатів досліджень компаній є звичайною справою. Це призупиняє дослідження, а також незважаючи на вже існуючи моделі АСК автомобілем, таку систему потрібно робити з нуля.

Для створення моделей стійкого сприйняття виявлення та сегментації використовують сучасні методи машинного навчання, але в даний час ведеться багато досліджень, спрямованих на підвищення надійності та продуктивності для досягнення можливостей людського рівня. Доступ до великих наборів даних має вирішальне значення для цих зусиль. Завдяки більшій кількості навчальних даних моделі сегментації і виявлення працюють краще і надійніше, але збір та маркування даних для всіх можливих типів транспортних засобів, погодних умов і дорожніх покриттів - дуже дорогий і трудомісткий процес. В даній дипломній роботі буде проведений порівняльний вибір моделей нейронних мереж для АСК автомобілем.

1.3 Методи та використовуванні технології для рішення питання побудови АСК автомобілем

Дослідження та розробка АСК автомобілем проводиться без фінансування та надання доріг для навчання і тестування системи, тому для рішення цих проблем як альтернатива був використаний симулятор CARLA. Він був вибраний, тому що цей симулятор призначений для навчання алгоритмів управління безпілотними автомобілями. Він дозволяє імітувати роботу різних сенсорів і отримувати дані в реальному часі. У ньому імітується міське середовище з будівлями, пішоходами, автомобілями та іншими об'єктами, а також змінюється погода. АСК

автомобілем була розроблена на прикладі моделі звичайного чотиримісного легкового автомобіля при правосторонньому русі.

При розробці з нуля для того, щоб скласти постановку задачі, в першу чергу необхідно визначити основні компоненти, терміни і концепції для створення безпілотного автомобіля, а також технічні вимоги, що лежать в основі їх конструкції і визначають самоуправління для автомобіля. Дане дослідження можна розділити на кілька блоків:

- таксономія безпілотних автомобілів або система класифікації, яку можна використовувати для визначення автоматизації водіння;

- вимоги для моделей сприйняття при водінні для визначення елементів, які необхідно правильно ідентифікувати при водінні;

- модель прийняття рішень при водінні, підходи до вибору того, як транспортний засіб буде рухатися в навколишньому середовищі.

Перший термін в списку - завдання водіння. Завдання водіння складається з трьох блоків. Перший блок - це сприйняття або визначення навколишнього середовища, в якому АСК автомобілем їде. Це включає в себе відстеження руху автомобіля для ідентифікації різних елементів навколишнього світу, таких як дорожнє покриття, дорожні знаки, автомобілі, пішоходів і так далі. Також необхідно відстежувати всі рухомі об'єкти і прогнозувати їх майбутній рух. Так щоб можна було їздити безпечно і акуратно. Другий блок - планування руху. Планування дозволяє успішно дістатися до місця призначення, наприклад, для того, щоб поїхати з дому до офісу. Таким чином, потрібно продумувати, якими шляхами слід рухатися, коли слід міняти смугу руху або перетинати перехрестя, а також як виконувати маневр з поворотом навколо вибоїни на шляху. Третій блок - система управління транспортним засобом, котрій потрібно приймати відповідні рішення при рульовому управлінні, гальмуванні і прискоренні, щоб контролювати стан і швидкість автомобіля на дорозі. Ці три блоки складають основну задачу водіння і повинні виконуватися постійно під час водіння автомобіля. Наступна концепція, необхідна для вирішення

завдання, називається домен операційного проектування або скорочено англійською ODD (Operational Design Domain). ODD це робочі умови, при яких дана система придатна для роботи. Вони включають в себе екологічні умови, час доби, проїжджу частину та інші характеристики, при яких автомобіль буде надійно працювати. Чітке визначення умов експлуатації, для яких розроблений безпілотний автомобіль, має вирішальне значення для забезпечення безпеки системи. Тому ODD потрібно ретельно спланувати заздалегідь. Для класифікації рівня автоматизації в системі водіння враховуються скільки уваги водія потрібно. Наприклад, може водій дивитися фільм, поки їде на роботу або йому потрібно весь час утримувати увагу на рульовому колесі. Також враховується скільки дій водія необхідно. Наприклад, чи потрібно йому рулити, машина піклується про швидкість або він теж її контролює, чи потрібно змінювати смугу руху або автомобіль може залишатися на поточній смузі і так далі. Згідно цим критеріям складається таксономія автономного водіння. Стандарти постійно розвиваються, але для цілей класифікації буде використана декомпозиція, запропонована товариством автомобільних інженерів в 2018 році [2].

Маючи даний стандарт, спосіб опису завдання водіння при підвищенні рівня автоматизації можна сформулювати наступним чином. По-перше, повинен бути бічний контроль, який відноситься до задачі рульового управління і бічного переміщення по дорозі. Поворот вліво, вправо, рух прямо або відстеження кривої і так далі. По-друге, поздовжній контроль. Це завдання, в якій контролюється положення або швидкість автомобіля на проїжджій частині за допомогою таких дій, як гальмування або прискорення. По-третє, повинно бути виявлення і реагування, яке спирається на об'єкти і події, або скорочено англійською OEDR (Object and Event Detection and Response). OEDR - це здатність виявляти об'єкти і події, які негайно впливають на завдання водіння, і відповідним чином змушують на них реагувати. По-четверте, планування. Оскільки негайне

реагування вже є частиною OEDR, планування в першу чергу пов'язано з довгостроковими і короткостроковими планами, необхідними для поїздки до місця призначення або виконання маневрів, таких як зміна нахилу і перетин перехресть.

За стандартом SAE J3 016 рівні автоматизації починаються з повного людського сприйняття, планування і контролю, цей рівень називається нульовим. На цьому рівні відсутня автоматизація водіння взагалі. Все робить водій. Якщо автономна система допомагає водієві, виконуючи завдання управління в поперечному або поздовжньому напрямку, тоді даний рівень можна вже назвати першим рівнем автономії. Адаптивний круїз-контроль - хороший приклад першого рівня. З адаптивним круїз-контролем або англійською ACC (Adaptive Cruise Control) система може контролювати швидкість автомобіля. Але для цього потрібно, щоб водій виконував рульове управління. Таким чином, система може виконувати поздовжнє управління, але для бокового управління потрібна людина. Системи допомоги при утриманні смуги руху також відносяться до першого рівня. У режимі допомоги при утриманні смуги руху система може допомогти водію залишатися в межах смуги руху і попередити його, коли він наближається до її кордонів. Сьогоднішні системи покладаються на візуальне виявлення кордонів смуги руху в поєднанні з бічним контролем центрування смуги руху. На другому рівні система виконує як завдання управління, так і поздовжнє і поперечне управління в певних сценаріях руху. Прикладами функцій другого рівня можуть бути: GM Super Cruise і Nissan Pro Pilot Assist. В даний час багато виробників автомобілів пропонують продукти автоматизації другого рівня, включаючи Mercedes, Audi, Tesla і Hyundai. На третьому рівні або рівні умовної автоматизації система може виконувати виявлення об'єктів і подій і відповідати на них певною мірою на додаток до завдань управління. Однак в разі відмови, управління повинно бути передано водієві. Ключова відмінність між другим і третім рівнями полягає в тому, що водієві не

потрібно звертати увагу в певних конкретних ситуаціях, так як транспортний засіб може вчасно попередити водія, коли йому треба втручатися. Це суперечливий рівень автоматизації, оскільки автономна система не завжди може знати, коли в ній стався збій. Прикладом систем третього рівня може бути Audi A Luxury Sedan, що представляє собою автоматизовану систему керування, яка може автоматично переміщатися в умовах повільного руху. На наступному рівні можуть бути лише високо автоматизовані транспортні засоби, у яких система здатна досягти стану мінімального ризику, якщо водій не втрутиться вчасно в разі виникнення надзвичайної ситуації. Системи четвертого рівня можуть впоратися з аварійними ситуаціями самостійно, але можуть попросити водіїв взяти на себе управління, щоб уникнути непотрібного з'їзду на узбіччя дороги. На донному рівні пасажири можуть перевірити свій телефон або подивитися фільм, знаючи, що система здатна впоратися з надзвичайними ситуаціями та забезпечити безпеку пасажирів. Проте, четвертий рівень як і інші допускає автономне водіння лише в обмеженому ODD. Станом на 2020 рік тільки Waymo розгорнула транспортні засоби для громадського транспорту з таким рівнем автономії. Автопарк Waymo може впоратися із завданням водіння тільки в певній географічній зоні і тільки з номінальним набором робочих умов, де не потрібна необхідність участі водія. На п'ятому рівні система повністю автономна, а її ODD необмежена. Це означає, що вона може працювати при будь-яких необхідних умов. П'ятий рівень - це точка, в якій суспільство зазнає трансформаційні зміни. Таксі без водія доставляють людей туди, куди їм треба. На даний момент поки немає прикладів п'ятого рівня, але, можливо, скоро цей рівень буде втілений в життя.

Наступним етапом необхідно розглянути вимоги до моделей сприйняття, які є ключовим аспектом проектування автономних систем. Грубо кажучи, будь-які завдання водіння можна розбити на дві складові. По-перше, потрібно розуміти, що відбувається навколо автомобіля і де він

знаходиться. Тобто, потрібно сприймати оточення. По-друге, потрібно приймати рішення по водінню. Зокрема, для будь-якого агента або елемента на дорозі потрібно спочатку визначити, що він таке; автомобіль, велосипедист, автобус і т. д. Далі необхідно зрозуміти його рух; рухається він певним чином, що може сказати, що він буде робити далі. Люди добре розуміють закономірності. Однак для комп'ютерних систем все ще важко розпізнавати ті ж самі шаблони навколо так швидко, як це робить людина. Людина може вказати на машину, що їхала прямо, і сказати: «О, вона буде в цьому положенні через деякий час в майбутньому». Прогнозування, це те, що робить можливим водіння. Отже, ця здатність прогнозувати траєкторію рухомого об'єкту дійсно важлива для сприйняття. Якщо можна зробити цей прогноз правильно, можна буде приймати обґрунтовані рішення. Наприклад, якщо я знаю, що машина переді мною збирається робити далі, я можу вирішити, що робити далі, таким чином, щоб обидві наші цілі були досягнуті. Для виконання даного процесу потрібно ідентифікувати різні елементи для завдання сприйняття. Потрібно визначити статичні елементи. Це такі елементи, як дороги і розмітка смуг, елементи, що розділяють ділянки на дорогах, наприклад переходи зебри, і важливі повідомлення, такі як школа попереду. Все це є на проїжджій частині. Потім є позашляхові елементи, такі як бордюри, які визначають межі, в межах яких можна проїхати. Є дорожні світлофори, які періодично змінюються і сигналізують, чи дозволено рухатися вперед, вліво, вправо або потрібно зупинитися. Крім того, існують різноманітні дорожні знаки, наприклад, які вказують на обмеження швидкості, напрямок, наближаємося до лікарні чи наближаємося і т. д. Знову ж таки, це позашляхові елементи. Нарешті, є дорожні перешкоди. Наприклад, помаранчеві конуси, які говорять, що йде будівництво, або що є межа блокпоста і так далі. Це також елементи дороги. Окрім статичних є динамічні елементи, які потрібно ідентифікувати для сприйняття. Це елементи, рух яких необхідно прогнозувати, щоб приймати обґрунтовані

рішення при водінні. Для цього необхідно ідентифікувати інші транспортні засоби на дорозі: чотириколісні автомобілі, такі як вантажівки, автобуси, автомобілі і т. д., а також необхідно ідентифікувати і прогнозувати рух двоколісних транспортних засобів, таких як мотоцикли, велосипеди і т. д. Останні рухомі системи мають більшу свободу, ніж чотириколісні автомобілі, тому їх важче передбачити. Нарешті, також потрібно вміти визначати і прогнозувати рух пішоходів навколо. Поведінка пішоходів сильно відрізняється від поведінки транспортних засобів, оскільки відомо, що пішоходи набагато більш нестабільні у виборі напрямку руху, ніж транспортні засоби, через властиву людям свободи в тому, як вони пересуваються. Ще одна важлива мета сприйняття - це локалізація. Потрібно мати можливість оцінити, де знаходиться автомобіль і як він рухається в будь-який момент часу. Знання положення і того, як рухається автомобіль в навколишньому середовищі, має вирішальне значення для прийняття усвідомлених і безпечних рішень при водінні. Дані, що використовуються для оцінки руху, надходять від датчиків GPS (Global Positioning System), IMU (Inertial Measurement Unit) і одометра, і їх необхідно об'єднати разом, щоб створити цілісну картину становища. Сприйняття не застраховане від невизначеності. У багатьох випадках видимість може бути складною, або вимірювання GPS спотворюються, або сигнали лідара і радара посилають лише шум замість значення місцеположення. Кожна підсистема, яка використовує ці датчики, повинна враховувати неточні вимірювання. Ось чому вкрай важливо розробити підсистеми, які можуть враховувати похибку датчика і спотворені вимірювання в кожній задачі сприйняття. У додаток є такі ефекти, як оклюзія і відображення в даних камери або LIDAR. Це може призвести до плутанини в методах сприйняття з неоднозначною інформацією, яку складно перетворити в точні оцінки місця розташування об'єктів. Існують також такі ефекти, як різка зміна освітленості і відблиски об'єктива або збої GPS і тунелі, які роблять деякі дані датчиків повністю непридатними

або недоступними. Методи сприйняття потребують декількох надлишкових джерелах інформації, щоб подолати втрату даних датчика. Нарешті, є погода і опади, які можуть негативно вплинути на якість вхідних даних з датчиків. Тому дуже важливо мати хоча б кілька датчиків, стійких до різних погодних умов, наприклад радар.

Планування можливо розбити на три види рішень. Перший тип - це довгострокове планувальне рішення. Наприклад, як дістатися з дому на роботу. Відповідаючи на це питання, складається план місії, план високого рівня для всього завдання водіння. Картографічні додатки, які використовуються сьогодні, можуть дати такі інструкції з водіння: якими шляхами рухатися, по яким смугах руху і так далі. Другий тип - це короткострокові рішення при водінні з такими питаннями, як: чи безпечно зараз міняти смугу руху або коли слід повернути ліворуч на перехресті. Водіння також вимагає негайних рішень або реакцію. Ці рішення включають контроль і планування траєкторії, а також відповіді на такі питання, як: слідувати смузі руху по цій звивистій дорозі або яке рульове управління слід застосувати, розганятися або гальмувати і якщо так, то наскільки. Припустимо, автомобіль наближається до перехрестя на шляху додому. На етапі довгострокового планування необхідно повернути ліворуч на цьому перехресті. Проміжними і короткостроковими рішеннями, які необхідно прийняти будуть наступні: припустимо, що перехрестя контролюється. Тобто є світлофори. Оскільки відбувається поворот наліво, необхідно визначити, чи потрібно міняти смугу. Потім, коли автомобіль наближається до цього перехрестя, потрібно вирішити знизити швидкість і робити це плавно, щоб пасажири не відчували дискомфорту. Потім автомобіль зупиняється прямо перед стоп-лінією, перед пішохідним переходом. Ці рішення про зміну смуги руху та місця зупинки є короткостроковими планувальними рішеннями. В додаток до них, також потрібно думати і реагувати на ситуації, які виникають в ході шляху. Для цього потрібно OEDR. Що робити, якщо перед автомобілем на

поворотну смугу виїжджає автомобіль? Слід зупинитися раніше, щоб звільнити місце для іншого транспортного засобу. Що, якби стоп-лінії не були зазначені? Потрібно буде приблизно оцінити, де мається на увазі стоп-лінія, і зупинитися перед пішохідним переходом. Що, якби позаду були інші автомобілі або навіть були б автомобілі, які зупинилися на перехресті? Яким чином рішення про зміну повороту наліво ґрунтується на безлічі можливих сценаріїв, які можуть швидко виникнути при нормальному водінні? Всі ці рішення відносяться до категорії негайних рішень і вимагають акуратної реакції з боку системи планування. Кінцевим результатом є великий список можливих рішень для оцінки в різних часових масштабах, навіть для простого сценарію, де автомобіль робить лівий поворот. Це зводиться до обговорення різних випадків для одного і того ж перетину перехрестя або сценарія. У кожному сценарії потрібен узгоджений набір варіантів, який буде оцінюватися в реальному часі і оновлюватися в міру появи нової інформації про середовище. Крім того, оскільки рішення про зміну смуги руху впливають на те, де їхати і які автомобілі регулюють положення щодо дороги, навіть простий сценарій водіння вимагає трьох або чотирьох рівнів рішень, які потім повинні виконуватися обережно. Цей приклад насправді лише поверхневий шар постійного потоку рішень, необхідних для планування руху. Для подібних сценаріїв необхідна структура для подання цих рішень до програмного забезпечення. Одним з методів вирішення проблеми багаторівневого прийняття рішень є реактивне планування. У реактивному плануванні визначається набір правил, які беруть до уваги поточний стан автомобіля і інших об'єктів в навколишньому середовищі і виробляють негайні дії. Іншими словами, це правила, які враховують тільки поточний стан, але не прогнози на майбутнє. Приклади таких правил: якщо на дорозі є пішохід, потрібно зупиниться. Або, якщо обмеження швидкості зміниться, потрібно відрегулювати швидкість відповідно до нього. В обох цих правилах спостерігається за тим, що відбувається прямо зараз, і приймається

рішення на основі щойно отриманої інформації. Але є й інші види планування. При прогнозному плануванні робляться прогнози щодо того, як інші агенти в навколишньому середовищі, такі як автомобілі і пішоходи, будуть рухатися з плином часу. Для цього використовується їх поточний стан і інформація про прогноз для визначення всіх рішень. Приклади правил прогнозного планування: автомобіль зупинився на останні 10 секунд, ймовірно, він буде стояти протягом наступних ще кількох секунд тому, можливо, є спосіб безпечно обійти його. Або пішохід підходить до переходу, коли автомобіль під'їде до переходу, пішохід уже буде переходити дорогу. Значить потрібно сповільнитися і дати йому можливість перейти дорогу. Як видно, це більш природний спосіб мислення, тісно пов'язаний з тим, як люди керують транспортними засобами. Вони прогнозують, де будуть інші об'єкти на дорозі в майбутньому, перш ніж приймати рішення. Однак цей тип планування ґрунтується на точному прогнозуванні дій інших суб'єктів у навколишньому середовищі, що значно ускладнює завдання сприйняття. Проте, прогнозне планування є переважаючим методом для безпілотних автомобілів, оскільки він значно розширює сценарії, з якими транспортний засіб може безпечно впоратися.

1.4 Аналіз напрямку розробки

1.4.1 Аналіз різних точок зору, тенденції при розробці

Існує 3 основних точки зору на тему того, як слід будувати систему автоматичного управління автомобілем:

- будувати систему автоматичного управління автомобілем, де незалежно від навколишнього середовища машина самостійно виконує всі функції водія, замінюючи його;

- будувати інфраструктуру міста, де машини будуть контролюватися системою міста та управління буде оптимізовано під дорожній трафік в цілому;

- будувати єдину систему для управління декількома автомобілями, наприклад, що належить одній компанії таксі або доставки в цілому.

Проаналізувавши вищезгадані варіанти створення системи для автоматичного керування автомобілем було прийнято рішення реалізовувати ідею першого варіанту з автономної машиною. Ухваленню цього рішення стала низка факторів.

Виникали питання: чи повинні самі транспортні засоби бути інтелектуальними або вони повинні використовувати інтелектуальну інфраструктуру? Що ж таке розумна інфраструктура? Світлофор, який повідомляє про своє місце розташування, свій поточний стан, а також те, що він збирається стати червоним, тощо. Є над чим подумати при реалізації різних способах створення інструментів для автономної інфраструктури міст. Тому я вважаю, що самі транспортні засоби повинні бути досить розумними, щоб працювати без залежності від інфраструктури. І якщо в місті вже буде розумна інфраструктура, то, звичайно, можна нею користуватися, за умови що їй можна буде довіряти. На даний момент, існує безліч питань щодо кібербезпеки, і це дійсно такі питання кібербезпеки, які зачіпають всі складові елементи цієї інфраструктури. Таким чином, я вважаю, що автомобіль повинен бути досить розумним, щоб керувати самостійно, як водій. Водій може дивитися на світлофор, може дивитися на межі смуги руху, і йому не потрібно, щоб машина попереду говорила, що вона попереду, водій сам може це бачити.

Також у другого і третього варіанту виникає проблема: якщо цю інфраструктуру прибрати, все одно буде хотітися мати автономний транспортний засіб для виїзду, наприклад, за місто. Воно може бути доповнено інфраструктурою, але я вважаю, що воно не повинно залежати від неї, тому що якщо йти за першим сценарієм, тоді у вас дійсно буде в

доступності весь можливий список автономного транспорту в будь-якому місці під рукою. Звичайно, для розробки цього може знадобитися деякий додатковий час, ніж у другого і третього варіантів. Але при цьому не треба буде залежати від дорогої інфраструктури, на яку щоб накопичити достатньо коштів може знадобитися ще більше часу, ніж щоб реалізувати перший варіант. Тому моя точка зору така, що дорогу, створену на замовлення інфраструктуру для безпілотних автомобілів варто надати як справу розвитку самого міста.

1.4.2 Аналіз ринку предметної галузі

За станом на сьогодні галузь автопілотів є широко рекламованою як система, котра покращить управління своїм часом, тому що не буде потребувати втручання людини при керуванні автомобілем. Перед рішенням поставленої задачі потрібно провести порівняльний аналіз впливу очікувань на розвиток галузі та дослідження.

У міру появи прототипів високоавтоматизованих транспортних засобів зростає і інтерес громадськості та медіа до можливих наслідків розповсюдження автопілотів. Переваги появи автопілотів включають підвищену безпеку, менше проблем з дорожнім рухом та додатковий робочий час чи дозвілля в транспортному засобі. Початкові звіти свідчать про те, що громадськість почала сприймати зображення автопілотів, в яких людина не має контролю як вимогу за замовчуванням.

Ці очікування можуть відображати надмірну впевненість у здатності автоматизувати водіння. Впровадження автономних транспортних засобів стикається із значними невирішеними проблемами. Технологія підтримки автопілоту залишається обмеженою, що перешкоджає широкому розгортанню автономних транспортних засобів. Лише нещодавно вирішення дилем на дорозі почало отримувати належну увагу.

Схожі приклади зроблених складних автоматизованих систем, наприклад, в авіації можуть пояснити чому потрібно зменшити людський фактор, на відміну від цілі усунення людини від керування автомобілем, яку може бути важко досягти і може бути навіть не бажаним результатом. Якщо автоматизація водіння не може бути реалізована з досконалою або майже ідеальною надійністю - результат, який здається неправдоподібним, особливо під час передбачуваних перехідних фаз розгортання, під час яких автопілоти будуть розділяти дороги з традиційними транспортними засобами - людина ймовірно вважатиме за краще наглядати під час автоматизованого водіння. Одним з добре відомих наслідків автоматизації є те, що вона має тенденцію перекладати навантаження з ручних завдань на завдання моніторингу, іноді без зменшення загального навантаження. Водіям також потрібно постійно підтримувати обізнаність про сценарій водіння в очікуванні дуже рідкісних випадків, коли буде необхідне втручання. Ці типи завдань пильності є особливо напруженими та стресовими, а із-за рідкості виконуються погано, тому що трапляються коли людина не готова прийняти керування на себе.

Навіть якщо відповідні інтерфейси можуть бути розроблені, щоб утримувати водіїв у курсі, залишається незрозумілим, чи приймуть споживачі автоматизований транспортний засіб, який міг би виконувати всі водійські завдання, виконував більшість водійських завдань, проте вимагав великого обсягу моніторингового навантаження. Таке навантаження може бути результатом навмисного проектного рішення про розподіл функцій контролю за водієм. Моніторинг також може виникнути через відсутність довіри водіїв до системи. Обидва сценарії можуть перешкодити або відмовити водія від участі у другорядних завданнях. Моніторинг робочого навантаження може негативно вплинути на прийняття системи на ринку, оскільки залучення до другорядних завдань розглядається як основна перевага кінцевого споживача від автоматизації транспортних засобів.

Крім того, дослідження показали, що початкові ідеалістичні очікування щодо автоматизації особливо шкідливі для подальшої довіри та прийняття, коли ці очікування не узгоджуються з функціональними можливостями людей, які користуються системою. Хоча остаточні можливості автопілотів ще далеко не реалізовані, нереальні очікування можуть стати перешкодою для прийняття. Високо ідеалізовані зображення почали формувати сподівання, що автопілоти вимагатимуть незначного чи взагалі відсутності людського втручання та створюватимуть додатковий час для роботи, відпочинку чи соціального часу під час поїздки. Сценарій, який здається однаково або, можливо, навіть більш правдоподібним, особливо на ранніх етапах розгортання, - це той варіант, коли водій уважно стежить за автоматизованими транспортними засобами і повинен бути готовим відновити ручне управління в короткі терміни протягом більшості або всього її часу.

Початкове розгортання автопілотів може сповільнитися або зашкодити, якщо технологія буде сприйнята розчаровано. На довіру до автоматизації впливають очікування та ставлення, які складаються до того, як людина використовує систему, тому буде важливо зрозуміти прийняття до реалізації автопілоту. У тій мірі, в якій ідеалізовані зображення автоматизації автомобілів вже почали впливати на прийняття, вони можуть також заохочувати нереальні очікування щодо ефективності автоматизації, які можуть мати зворотний ефект для прийняття в довгостроковій перспективі.

Щоб зрозуміти на якому зараз рівні прийняття суспільством автопілоту було проведено дослідження Lafayette коледжем. Результати якого можна побачити в таблиці 1.1.

Таблиця 1.1 - Результати SCAS (Self-driving Car Acceptance Scale)

Предмет дослідження	M	SD	Mdn	Mode
Отримана довіра				
1. Автопілот безпечний	5.08	1.27	5	6
2. Я довірю автопілоту довести мене до місця призначення	5.01	1.46	5	6
3. Водієві необхідно уважно спостерігати за роботою автопілота	5.09	1.42	5	5
Ціна				
4. Я готовий заплатити більше за автопілот у машині	4.28	1.88	5	6
5. Переваги автопілотів перевищують їх ціну	4.30	1.73	4	6
6. Ціна автопілота найважливіша частина перед покупкою	4.76	1.72	5	6
Доцільність автоматизації / сумісності				
7. Я не думаю, що комп'ютери повинні керувати автомобілями	3.33	1.72	3	2
8. Для людини важливо мати можливість повернути управління водієві	6.17	1.03	6	7
9. Є кілька сценаріїв водіння, які буде надто складними для автопілоту	5.02	1.59	5	6
Задоволення автоматизацією				
10. Мені подобається водити машину	5.33	1.47	6	6
11. Я вважаю за краще бути водієм, а не пасажиром в машині	4.55	1.86	5	6
12. Мені подобається подорожувати круїзом	5.07	1.62	5.50	6
Сприйнята корисності автоматизації				
13. Автопілот дозволив би мені бути більш продуктивним	5.18	1.59	6	6
14. Автопілот дозволив би мені бути в безпеці, перебуваючи в машині	4.82	1.46	5	6
15. Автопілот зменшить проблеми з дорожнім рухом	4.93	1.57	5	6
Сприймається простоти використання засобів автоматизації				
16. Автопілотами буде просто користуватися	5.24	1.28	5	6
17. Мені знадобилося б багато часу, щоб зрозуміти	3.14	1.61	3	2
Досвід роботи з автоматизацією				
18. Я люблю використовувати технології, що полегшують життя	6.09	0.86	6	6
19. У мене поганий досвід використання нових технологій	2.35	1.29	2	2
20. У моєму житті є завдання, які полегшили комп'ютери	6.37	0.76	6	7
Намір використовувати автоматизацію				
21. Я хотів би мати власний автопілот	4.91	1.84	5	6
22. Навіть якби у мене був автопілот, я все одно хотів би їздити самостійно	4.29	1.78	5	5
23. У автопілоті мені важливо мати можливість інколи їздити самостійно	6.20	0.96	6	7

У цьому експерименті учасники читають або реалістичний, або ідеалізований опис досвіду близького друга або члена сім'ї протягом перших шести місяців володіння автономним автомобілем. Даний тест має 23 питання та їх оцінку (1-7), M - математичне очікування, SD (Standard Deviation) - стандартне відхилення, Mdn - медіана та Mode - мода.

Експеримент показав, що люди більше сприймають автопілот після прочитання з ідеалізованим зображенням досконалого автопілоту порівняно як із станом управління, так і зі сценарієм, що описує більш реалістичний сценарій, коли водій-людина відігравав моніторингову роль і час від часу втручався під час автоматизації транспортних засобів. Ідеалізовані зображення можуть створити нереальні очікування щодо

продуктивності самохідних автомобілів. Прийняття автопілотів може зашкодити, якщо люди повинні виконувати моніторингові завдання під час автоматизації, коли вони очікують інше. Дослідження людських факторів показали, що нереальні очікування щодо ефективності автоматизації дійсно збільшують початкове прийняття, але довіра та прийняття можуть бути непоправно зашкоджені після того, як взаємодія з недосконалою автоматизацією виявить її нерозкриті раніше недоліки для користувачів. Це означає, що при реалізації автопілоту потрібно з особливою увагою поставитися до зменшення моніторингу водієм трафіку на дорозі при автопілоті.

1.5 Постановка завдання

У ході проведеного аналізу предметної галузі та можливостей для розробки був окреслений наступний список пропозицій для постановки завдання: проаналізувати та дослідити існуючі алгоритми управління АСК автомобілем, вибрати датасети для навчання нейронних мереж, проаналізувати та дослідити існуючі нейронні мережі для рішення питань сегментації, класифікації, сприйняття дорожніх об’єктів, розробити модель АСК автомобілем, визначити математичні моделі підсистем і елементів системи управління автомобілем, реалізувати АСК автомобілем, провести тестування отриманої моделі і зробити висновки на базі отриманих результатів.

В якості предмету дослідження АСК автомобілем був використаний симулятор CARLA. Дослідження результатів буде проходити в імітованому міському середовищі з будівлями, пішоходами, автомобілями та іншими об'єктами, а також зі змінною погодою. Автомобіль для досліджень - модель звичайного чотиримісного легкового автомобіля. При управлінні будуть використанні правила правостороннього руху.

2 ДОСЛІДЖЕННЯ АЛГОРИТМІВ УПРАВЛІННЯ, ДЕТЕКЦІЇ, КЛАСИФІКАЦІЇ, ПЛАНУВАННЯ МАРШРУТУ

2.1 Аналіз та дослідження існуючих алгоритмів управління АСК автомобілем

В ході дослідження алгоритмів управління були розглянуті наступні алгоритми управління: П, ПІ та ПІД-регулятори, лінійні та нелінійні регулятори, алгоритм Калмана та його варіації, такі як розширений алгоритм Калмана та управління по помилці.

Пропорційний-інтегрально-диференційний (ПІД) регулятор - пристрій в керуючому контурі зі зворотним зв'язком. Використовується в системах автоматичного управління для формування керуючого сигналу з метою отримання необхідної точності і якості перехідного процесу. ПІД-регулятор формує керуючий сигнал, який є сумою трьох доданків, перший з яких пропорційна різниця вхідного сигналу і сигналу зворотного зв'язку (сигнал неузгодженості), другий - інтеграл сигналу неузгодженості, третій - похідна сигналу неузгодженості. Якщо якісь із складових не використовуються, то регулятор називають пропорційно-інтегруючим, пропорційно-диференційним, пропорційним і т. д. Загальну схему ПІД-регулятора можливо побачити на рисунку 2.1.

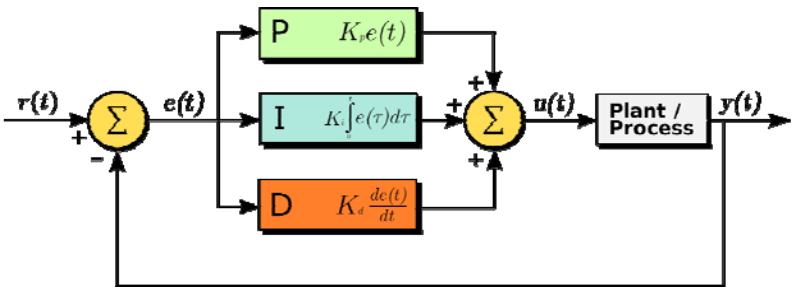

Рис. 2.1 – Загальна схема ПІД-регулятора

Ця система управління зі зворотним зв'язком за участю ПІД-регулятора управляє величиною y(t), тобто виводить величину y(t) на задане ззовні значення r(t). На вхід ПІД-регулятора подається помилка e(t), а вихід ПІД-регулятора є керуючим впливом u(t) для деякого процесу (для об'єкта управління), керуючого величиною y(t).

На жаль ПІД-регулятори мають недоліки при їх використанні на практиці. При використанні ПІД-регулятора в системі регулювання, слід враховувати небажані ефекти, що виникають при реалізації каналу похідної сигналу помилки ė(t). Недоліки проявляються через те, що при посиленні цього каналу прямо пропорційно зростає частота. Основними недоліками при цьому є: підвищене посилення високочастотних складових сигналу помилки. Вони носять шумовий характер і через це відношення корисної складової сигналу до шумовий зменшується, що дестабілізує об'єкт управління. Також може виникати імпульси великої амплітуди. Таке явище виникає в моменти стрибкоподібної зміни помилки.

Стандартним прикладом використання ПІД-регулятора в даній предметній галузі є круїз-контроль автомобіля: на виході відображається кількість палива, що відправляється в двигун. Датчик спідометр. Вимірювання спідометра порівнюється з бажаною швидкістю, і вихідний сигнал налаштовується. Це дозволяє машині регулювати кількість газу, коли машина піднімається і опускається по схилах.

Хоча це добре працює для багатьох пристроїв, більш складні системи мають ряд керованих виходів, а також повинні мати справу з гучними входами датчиків. Тому для управління було вирішено, що більш підійде фільтр Калмана або його похідні. Даний алгоритм часто використовується для систем наведення. Основна ідея полягає в тому, що фільтр Калмана відстежує стан системи, має модель роботи системи і робить прогнози на основі цієї моделі. Прогнозований вихідний сигнал потім можна порівняти з новими показаннями датчика, і знання про стан оновлюються.

Так що, оскільки АСК автомобілем не прив'язана до зовнішнього контролю, можливо використовувати фільтр Калмана, щоб допомогти їй орієнтуватися. Фільтр починається з відомого стану автомобіля: зупинений, колесо направлено, наприклад, на північ, а також на широту X і довжину Y. Щоб досягти нового пункту призначення, фільтр може генерувати вхідні дані (газ і кут повороту колеса), робити прогнози на основі фізичної моделі автомобіля (якщо повернути колесо на 90 градусів на швидкостях шосе, відбудуться погані речі), брати сенсорні входи (компас, спідометр, датчик пробігу) і, нарешті, подивитися, чи достатньо близькі ці прогнози до реального середовища. Далі ця робота повторюється.

Фільтр Калмана - це фільтр, який працює як оптимізатор мінімальних квадратних помилок і для цього необхідно, щоб система, всередині фільтра, була лінійною.

Для того, щоб зробити оцінку стану в нелінійних системах або оцінку параметрів, використовуючи фільтр Калмана, одним з можливих підходів є лінеаризація досліджуваної системи навколо її поточного стану та примушення фільтра використовувати цю лінеаризовану версію системи як модель. Дана варіація називається розширеним фільтром Калмана, або EKF.

Однак EKF не дуже стабільний, і багато разів, коли він наближається до "правильного" рішення, він робить це дуже повільно. Для того, щоб вдосконалити цей фільтр, в кінцевому результаті в АСК автомобілем був використаний ES-EKF (Error State Extended Kalman Filter) - розширений фільтр Калмана за станом помилки.

Принцип роботи ES-EKF в АСК автомобілем можливо писати формулою 2.1.

$$\mathbf{x} = \hat{\mathbf{x}} + \delta\mathbf{x} \tag{2.1}$$

де x - справжнє значення;

x^ - номінальне значення;

δx - значення помилки.

Роботу ES-EKF можливо побачити на прикладі відстеження положення автомобіля з часом показаному на рисунку 2.2.

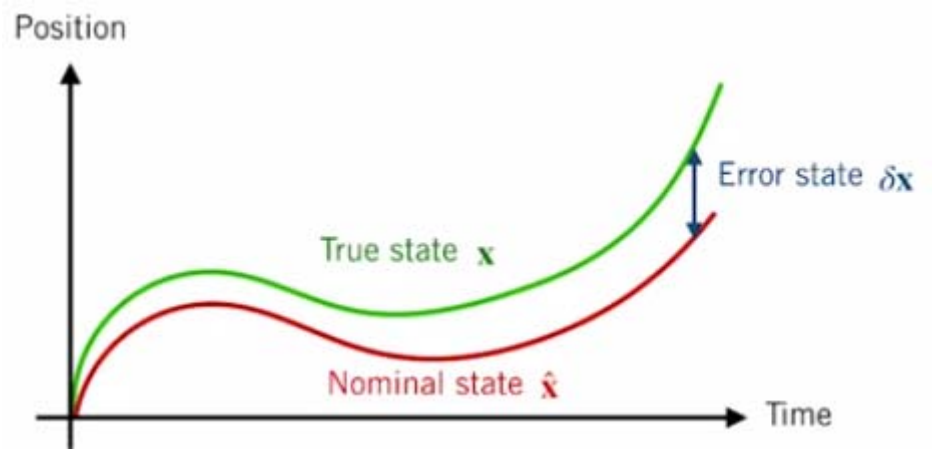

Рис. 2.2 - Відстеження положення автомобіля з часом

На рисунку зелена лінія показує справжнє положення автомобіля, тобто кількість, яку потрібно оцінити. Червона лінія - це номінальний стан, або найкращий прогноз, яким може бути справжній стан, базуючись на тому, що відомо про модель руху автомобіля, а також дані про прискорення та розриви. Модель руху ніколи не буває ідеальною, і завжди виникають випадкові шуми процесу. Ці помилки накопичуються з часом, коли йде інтеграція моделі руху. Стан помилок це місце, де всі помилки моделювання та шум процесу накопичуються з часом, таким чином стан помилки - це різниця між номінальним станом та справжнім станом у будь-який момент часу. Якщо з'ясувати, який стан помилок, можливо використовувати його як корекцію номінального стану, щоб наблизитись до справжнього стану. Отже, у стані помилки EKF, замість фільтрації

Калмана на повний стан, який може мати багато складної нелінійної поведінки, використовується EKF для оцінки стану помилки, а потім використовується оцінка стану помилки як поправка на номінальний стан. Математично це означає, що лінеаризована модель руху переставляється так, що вийде рівняння, яке може сказати, як пов’язана різниця між справжнім станом в момент часу k, та прогнозованим станом у у часі k до тієї ж різниці в часі k-1. Рівняння, що їх пов'язують, називаються кінематикою стану помилок і показані в формулі 2.2.

$$\underbrace{\mathbf{x}_k - \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0})}_{\delta \mathbf{x}_k} = \mathbf{F}_{k-1} \underbrace{\left(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}\right)}_{\delta \mathbf{x}_{k-1}} + \mathbf{L}_{k-1}\mathbf{w}_{k-1} \tag{2.2}$$

Також можливо повторно виразити лінеаризовану модель вимірювання безпосередньо через стан похибки за формулою 2.3.

$$\mathbf{y}_k = \mathbf{h}_k(\check{\mathbf{x}}_k, \mathbf{0}) + \mathbf{H}_k \underbrace{\left(\mathbf{x}_k - \check{\mathbf{x}}_k\right)}_{\delta \mathbf{x}_k} + \mathbf{M}_k \mathbf{v}_k \tag{2.3}$$

Якщо тепер все записати у вигляді циклу, ES-EKF буде працювати наступним чином: спочатку оновлюється номінальни
й стан за допомогою нелінійної моделі руху та нашої найкращої оцінки стану за формулою 2.4.

$$\check{\mathbf{x}}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \tag{2.4}$$

This could be $\check{\mathbf{x}}_{k-1}$ or $\hat{\mathbf{x}}_{k-1}$

На цьому етапі найкращою оцінкою може бути $\hat{x}$. Також потрібно відслідковувати коваріацію стану за формулою 2.5, яка зростає, коли інтегрується все більше і більше процесного шуму з моделі руху.

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^T \tag{2.5}$$

This could be $\check{\mathbf{P}}_{k-1}$ or $\hat{\mathbf{P}}_{k-1}$

Далі потрібно обчислити коефіцієнт підсилення Калмана, а потім обчислити найкращу оцінку стану похибки за формулами 2.6, використовуючи коефіцієнт підсилення Калмана, вимірювання та нелінійну модель вимірювання.

$$\begin{aligned} \mathbf{K}_k &= \check{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\check{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R})^{-1} \\ \delta\hat{\mathbf{x}}_k &= \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}_k(\check{\mathbf{x}}_k, \mathbf{0})) \end{aligned} \tag{2.6}$$

Отримавши оцінку середнього стану помилки, вона використовується для оновлення номінального стану та виправлення помилки. Це можливо зробити додавши оцінку стану помилки до номінального стану, щоб отримати правильну оцінку для повного стану. В кінці роботи алгоритму оновлюється коваріація стану за формулою 2.1. Цей процес триває вічно, або до тих пір, поки в транспортному засобі не закінчиться бензин.

Причини чому в даній дипломній роботі було вибрано використовувати саме ES-EKF дві: однією з причин є те, що він часто може працювати краще, ніж звичайний EKF, оскільки стан малих помилок більше піддається лінійній фільтрації, ніж великий номінальний стан, який можливо інтегрувати не лінійно. Інша причина полягає в тому, що

формулювання стану помилки значно полегшує роботу з обмеженими величинами, такими як обертання.

2.2 Моделювання зовнішнього середовища та обрання навчаючого датасету

Дане дослідження могло піти двома напрямками. Ці напрямки напряму залежать від типу вхідних даних: збір вхідних даних на отриманій базі своїх датчиків, сенсорів прикріплених до машини або використання відкритих датасетів.

Кожен з напрямків має свої плюси та мінуси. Перевага першого напрямку в тому, що є можливість отримати багато даних реальних ситуацій на дорозі, котрі можуть покращити систему прийняття рішень, такі як пошкодження частини датчиків або погана погода. Але для цього потребує отримання ліцензії для можливості їздити автопілотом по громадським дорогам, наявність необхідного фізичного обладнання, такого як: машина, сенсори, датчики, комп’ютер для сканування ситуації, контролю руху та прийняття рішень і тому подібне. Другий варіант набагато дешевший, але не має тієї переваги, що є в першого.

Оскільки перший варіант передбачає велике додаткове фінансове вкладення капіталу у проект, яке на даному етапі не є можливим, було прийнято рішення піти вторим варіантом.

Для даного дослідження був використаний датасет від компанії KITTI Vision. Цей датасет складається з 3Д і 2Д анотацій. У ньому представлений великомасштабний набір даних, який містить багату сенсорну інформацію і повні анотації. У датасеті записано кілька передмість Карлсруе, Німеччина, що відповідає більш 320 тис. зображень і 100 тис. лазерних сканувань. Анотовані як статичні, так і динамічні елементи тривимірної сцени грубими обмежуючими примітивами, які передають інформацію в область зображення, в результаті чого виходять

семантичні екземпляри анотацій як для тривимірних хмар точок, так і для двомірних зображень. Дальність поїздки: 73,7 км, кадрів: 4 x 83000. Всі кадри точно геолокалізовани за допомогою OpenStreetMap. Семантичні мітки визначені відповідно до Cityscapes, а саме 19 класів для оцінки, де для кожного примірника призначається постійний ідентифікатор екземпляра у всіх кадрах.

Розташування сенсорів, а також їх результат можна побачити на рисунках 2.3-2.5.

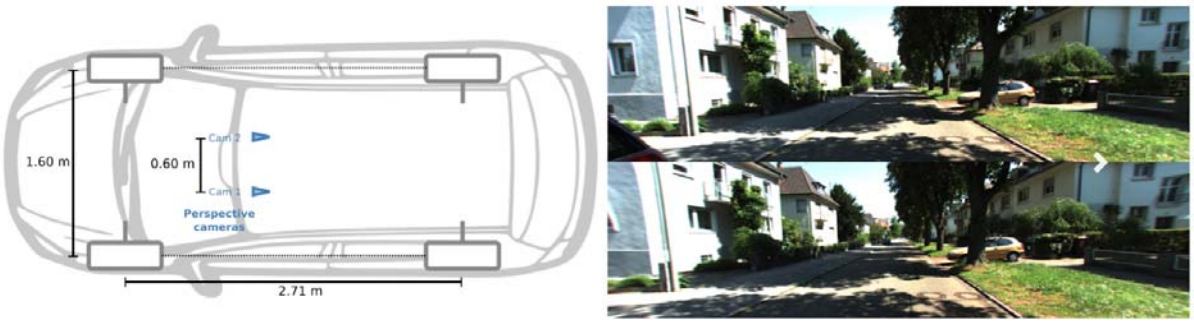

Рисунок 2.3 - Схема розташування сенсорів зліва, а також їх результат роботи справа

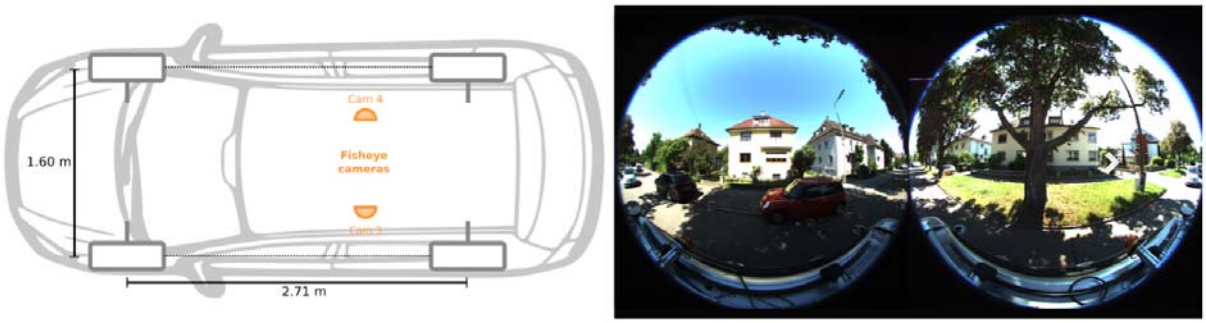

Рисунок 2.4 - Схема розташування сенсорів зліва, а також їх результат роботи справа

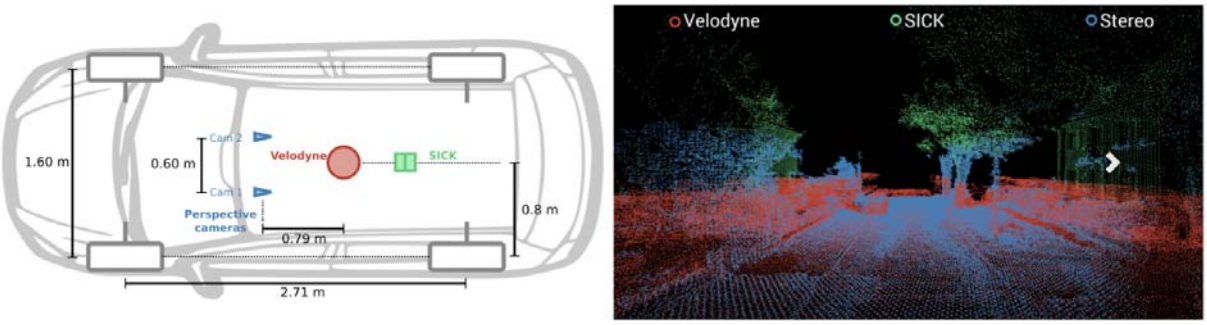

Рисунок 2.5 - Схема розташування сенсорів зліва, а також їх результат роботи справа

Як видно на рисунках вище для збору даних було обладнано по одній камері «риб'яче око» на 180 ° з кожного боку і стереокамерами з перспективою 90 ° (базова лінія 60 см) спереду. Крім того, було встановлено на даху Velodyne HDL-64E і SICK LMS 200 лазерний скануючий блок в конфігурації з ручками. Таким чином можна отримати поле огляду на 360 °. Крім того, система також оснащена системою локалізації IMU / GPS.

Це все дозволило отримати набір даних, приклади яких можна побачити на рисунку 2.6.

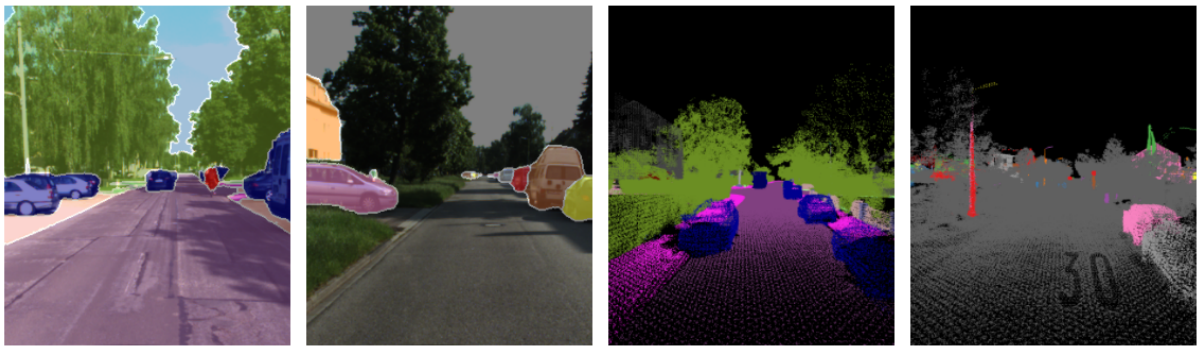

Рисунок 2.6 - Приклади з датасета KITTI Vision

Таким чином навчання моделей системи буде проходити за напрямком наявних доступних даних.

2.3 Аналіз та дослідження існуючих нейронних мереж для рішення питань сегментації, класифікації та сприйняття дорожніх об’єктів

В даній дипломній роботі було вирішено використовувати моделі DeepMask, Sharpmask та MultiPathNet. Вони були розроблені командою дослідників штучного інтелекту Facebook (FAIR) для ідентифікації та сегментування кожного об’єкта на входах зображень для використання в численних програмах глибокого навчання для виявлення об’єктів. Кожен із них виконує різні цілі в процесі. DeepMask і Sharpmask виступають як «очі» алгоритму, а MultiPathNet як «мозок». DeepMask - може знаходити об'єкти у вхідних зображеннях, але не може описувати їх та їх межі. Sharpmask - вдосконалює результати роботи DeepMask, додаючи маски вищої точності, що покращує точність виявлення об'єктів та їх меж. MultiPathNet - бере вихідні дані DeepMask та Sharpmask та класифікує їх.

Якщо сприймати ці алгоритми як людину, яка дивиться в небо і бачить об’єкт, тоді у цьому сценарії DeepMask схожий на людину з неозброєними очима. Вони можуть помітити об’єкт, але не можуть його ідентифікувати. Sharpmask схожий на телескоп, за допомогою якого вони можуть ідентифікувати об’єкт як птаха, а MultiPathNet служить орієнтиром, за яким вони можуть класифікувати птахів, яких бачать. Таким чином, замість того, щоб сказати «є об’єкт на небі», вони можуть надати набагато точніший опис: «це альбатрос». Вибір нейронної мережі DeepMask був зроблений на основі результатів досліджень моделей сегментації, а моделі детекції та класифікації об’єктів були вибрані опираючись вже на DeepMask.

При виборі нейронних мереж для рішення питань сегментації, класифікації та сприйняття дорожніх об’єктів було досліджено найновішу літературу та протестовані більше сотні методів на основі навчання, запропонованих до 2020 року. Для цього був зроблений всебічний огляд

різних аспектів цих методів, включаючи дані про навчання, вибір архітектури мережі, функції втрат, стратегії навчання та їх ключовий внесок. Результатами цього дослідження є порівняльний підсумок ефективності розглянутих методів. [18]

Моделі нейронних мереж, засновані на глибокому навчанні, були розділені на наступні категорії на основі їх основних технічних характеристик:

1) Повністю згорткові мережі;
2) Конволюційні моделі з графічними моделями;
3) Моделі на основі кодера-декодера;
4) Багатомасштабні та пірамідальні мережеві моделі;
5) Моделі на основі R-CNN (наприклад, для інстанс сегментації);
6) Розширені згорткові моделі та сімейство DeepLab;
7) Повторювані моделі на основі нейронних мереж;
8) Моделі, орієнтовані на увагу;
9) Генеративні моделі та змагальний тренінг;
10) Конволюційні моделі з активними контурними моделями;
11) Інші моделі.

2.3.1 Огляд глибинних нейронних мереж

Convolutional Neural Networks (CNNs) - належать до найбільш успішних та широко використовуваних архітектур у спільноті глибокого навчання, особливо для завдань комп’ютерного зору. CNN були спочатку запропоновані Фукусімою у його основній праці про "Неокогнітрон", заснованій на ієрархічній моделі рецептивного поля зорової кори, запропонованій Хубелем та Візелем. Згодом Вайбель представив CNN з вагами, розподіленими між тимчасовими рецептивними полями та навчанням зворотного розмноження для розпізнавання фонеми, а LeCun

розробив архітектуру CNN для розпізнавання документів, архітектура котрої показана на рисунку 2.7.

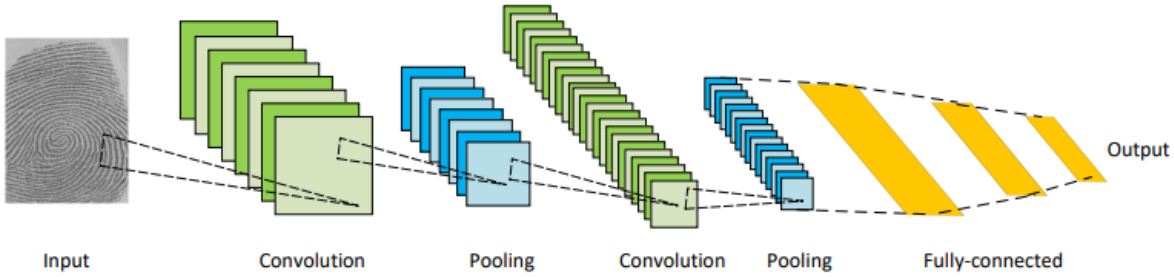

Рис. 2.7 - Архітектура CNN

CNN в основному складаються з трьох типів слоїв: згорткових слоїв, де ядро (або фільтр) ваг трансформується для того, щоб виділити ознаки; нелінійні слої, які застосовують функцію активації на картах об’єктів (як правило, поелементно), щоб забезпечити моделювання нелінійних функцій мережею, об'єднання слоїв, які замінюють невелике сусідство на карті об'єктів деякою статистичною інформацією (середнє, максимальне тощо) та зменшують просторову роздільну здатність. Одиниці у слоїв пов'язані локально; тобто кожна одиниця отримує зважені вхідні дані з невеликого сусідства, відомого як сприйнятливе поле одиниць попереднього слою. Складаючи слої для формування пірамід з роздільною здатністю, слої вищого рівня вивчають особливості з дедалі ширших сприйнятливих полів. Основна обчислювальна перевага CNN полягає в тому, що всі сприйнятливі поля в слої мають спільні ваги, що призводить до значно меншої кількості параметрів, ніж повністю підключені нейронні мережі. Серед найбільш відомих архітектур CNN є: AlexNet, VGGNet, ResNet, GoogLeNet, MobileNet і DenseNet.

Recurrent Neural Networks (RNNs) - широко використовуються для обробки послідовних даних, таких як мова, текст, відео та часові ряди, де дані в будь-який момент часу або позиції залежать від даних, які раніше

зустрічалися. На кожній мітці часу модель збирає вхідні дані поточного часу X_i та прихований стан з попереднього кроку h_{i-1}, і видає цільове значення та новий прихований стан. Архітектуру можливо побачити на рисунку 2.8.

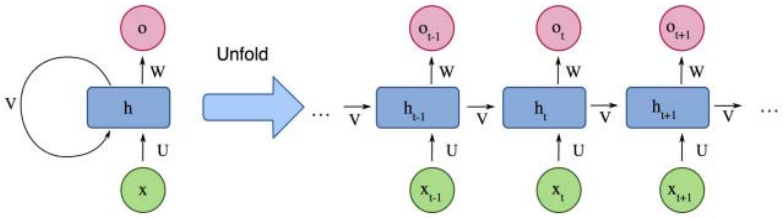

Рис 2.8 - Архітектура класичної RNNs моделі

RNN, як правило, проблематичні з довгими послідовностями, оскільки вони не можуть захопити довгострокові залежності у багатьох реальних програмах (хоча вони не мають теоретичних обмежень у цьому відношенні) і часто страждають від проблем зникнення або вибуху градієнта. Однак тип RNN, який називається Long Short Term Memory (LSTM), призначений для уникнення цих проблем. Архітектура LSTM (рис. 2.9) включає три ворота (вхідні ворота, вихідні ворота, забутні ворота), які регулюють потік інформації із комірки пам'яті, яка зберігає значення через довільні інтервали часу.

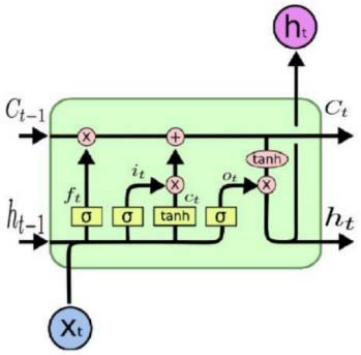

Рис. 2.9 - Архітектура LSTM

Зв'язок між входом, прихованими станами та різними воротами задається за формулою:

$$
\begin{aligned}
f_t &= \sigma(\mathbf{W}^{(f)} x_t + \mathbf{U}^{(f)} h_{t-1} + b^{(f)}), \\
i_t &= \sigma(\mathbf{W}^{(i)} x_t + \mathbf{U}^{(i)} h_{t-1} + b^{(i)}), \\
o_t &= \sigma(\mathbf{W}^{(o)} x_t + \mathbf{U}^{(o)} h_{t-1} + b^{(o)}), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(\mathbf{W}^{(c)} x_t + \mathbf{U}^{(c)} h_{t-1} + b^{(c)}), \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{2.7}
$$

де $x_t \in R^d$ - вхідний сигнал на кроці часу t;

d - розмірність ознаки для кожного слова;

σ - елемент сигмоїдної функції (для значень у межах [0, 1]);

$\odot$ - елементний добуток;

c_t - клітинка пам'яті, призначена для зниження ризику зникнення або вибуху градієнта;

f_t - вхід забуття призначений для скидання комірки пам'яті;

i_t - вхідні ворота, котрі керують входом комірки пам'яті;

o_t - вихідні ворота котрі керують виходом комірки пам'яті.

Моделі кодера-декодера та автоматичного кодування - це сімейство моделей, які вчаться зіставляти точки даних із вхідного домену у вихідний домен через двоступінчасту мережу: кодер, представлений функцією кодування z = f(x), стискає вхідні дані в подання в прихованому просторі; декодер, y = g(z), який має на меті передбачити вихідні дані із представлення прихованого простору. Приховане представлення відноситься до представлення ознаки (вектора), яке здатне захоплювати основну семантичну інформацію вхідного матеріалу, що корисно для прогнозування результату. Ці моделі надзвичайно популярні в задачах перекладу зображення на зображення, а також для моделей послідовностей в NLP. Рисунок 2.10 ілюструє блок-схему простої моделі кодера-декодера. Ці моделі, як правило, навчаються шляхом мінімізації втрат на

реконструкцію L(y, yˆ), що вимірює різницю між вихідним значенням істинної основи y та подальшою реконструкцією yˆ. Результатом роботи може бути покращена версія зображення або карта сегментації.

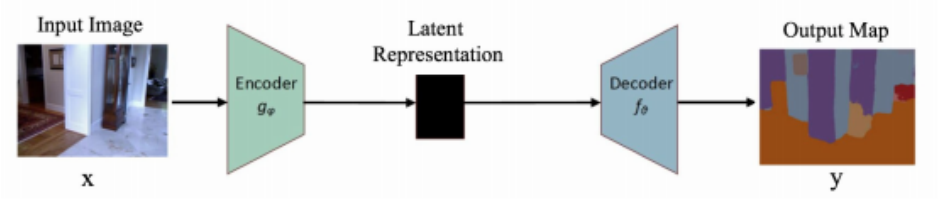

Рис. 2.10 - Блок-схема простої моделі кодера-декодера.

Автокодери - це випадок моделей кодерів-декодерів, в яких вхід і вихід однакові. Є кілька варіантів автоматичних кодерів. Одним з найпопулярніших є автоматичний кодер шумозаглушення (SDAE), який складає кілька автокодерів та використовує їх для цілей шумозаглушення. Іншим популярним варіантом є варіаційний автоматичний кодер (VAE), який накладає попередній розподіл на приховане представлення. VAE здатний генерувати реалістичні вибірки з заданого розподілу даних. Іншим варіантом є змагальні автоматичні кодери, які вносять змагальні втрати на прихованому поданні, щоб заохотити їх наблизити попередній розподіл.

Generative Adversarial Networks (GANs) - це нове сімейство моделей глибокого навчання. Вони складаються з двох мереж - генератора та дискримінатора (рис. 2.11).

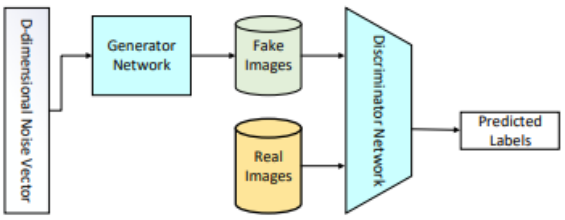

Рис. 2.11 - Архітектура GAN

Мережа генераторів у класичному GAN вивчає різницю відображення від шуму z (з попереднім розподілом) до цільового розподілу y, подібного до «реальних» вибірок. Мережа дискримінаторів D намагається відрізнити сформовані зразки «підробки» від «реальних». Функція втрати GAN може бути записано за формулою:

$$L_{GAN} = E_{x\sim pdata(x)}[\log D(x)] + E_{z\sim pz\,(z)}[\log (1 - D(G(z)))]] \qquad (2.8)$$

Можливо розглядати GAN як мінімаксну гру між G і D, де D намагається мінімізувати свою класифікаційну помилку, щоб відрізнити підроблені вибірки від реальних, отже, максимізуючи функцію втрат, а G намагається максимізувати помилку мережі дискримінатора, отже мінімізувати функцію втрат. Після навчання модель тренованого генератора буде $G^* = \arg \min_G \max_D L_{GAN}$. На практиці ця функція може не забезпечити достатнього градієнта для ефективної підготовки G, особливо спочатку (коли D може легко відрізнити фальшиві зразки від реальних). Замість того, щоб мінімізувати $E_{z\sim pz\,(z)}$ [log(1-D(G(z)))], можливе рішення полягає в тому, щоб навчити його максимізувати $E_{z\sim pz\,(z)}$ [log (D (G (z))))].

З часу винаходу GANs дослідники намагалися вдосконалити або модифікувати GANs декількома способами. Наприклад, Radford запропонував згорнуту модель GAN, яка працює краще, ніж повністю підключені мережі, коли використовується для генерації зображень. Mirza запропонував умовну модель GAN, яка може генерувати зображення, котрі обумовлені мітками класів, що дозволяє генерувати зразки із зазначеними мітками. Арджовський запропонував нову функцію втрат, засновану на Wasserstein (вона ж відстань землекопа), щоб краще оцінити відстань у випадках, коли розподіл реальних та генерованих зразків не перекриваються.

Трансферне навчання - у деяких випадках моделі можна навчити з нуля на нових додатках або наборах даних (припускаючи достатню

кількість мічених навчальних даних), але в багатьох випадках недостатньо мічених даних для навчання моделі з нуля, і можна використовувати трансферне навчання щоб вирішувати цю проблему. У процесі трансферного навчання модель, навчена вирішувати одне завдання, переробляється на інше (пов’язане) завдання, як правило, шляхом певного процесу адаптації до нового завдання. Наприклад, можна уявити, як адаптувати модель класифікації зображень, навчену в ImageNet, до іншого завдання, такого як класифікація текстур або розпізнавання обличчя. У випадку сегментації зображень багато людей використовують модель, навчену на ImageNet (датасет зображень), як кодерну частину мережі, і повторно навчають свою модель з початкових ваг. Тут передбачається, що ці попередньо навчені моделі повинні мати можливість фіксувати семантичну інформацію зображення, необхідну для сегментації, і, отже, даючи їм можливість тренувати модель з менш маркованими зразками.

2.3.2 Дослідження продуктивності нейронних мереж

Були використанні наступні метрики для дослідження продуктивності: IoU (перетин та об’єднання), Mean-IoU (середне значення перетину та об’єднання).

IoU - визначається як зона перетину між передбачуваною картою сегментації та основною істиною, котра розділена площею об'єднання між передбаченою картою сегментації та основною істиною за формулою:

$$\mathrm{IoU} = J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{2.9}$$

де A - основна істина;

B - передбачувані карти сегментації.

Mean-IoU - визначається як середнє значення IoU для всіх класів. Використовуючи дані метрики на вищеописаних нейронних мережах були отримані наступні результати:

Таблиця 2.1 - Результати за mIoU на датасеті KITTI Vision

Модель	Преднавчання	mIoU	Модель	Преднавчання	mIoU
DeepMask	2xResNet-50	85.5	DenseASPP	DenseNet-161	80.6
PSPNet	ResNet-101	85.4	PSANet	ResNet-101	80.1
HRNetV2+OCR (w/ASPP)	HRNetV2-W48	83.7	DFN	ResNet-101	79.3
GS-CNN	WideResNet	82.8	BiSeNet	ResNet-101	78.9
OCR	ResNet-101	82.4	Wide ResNet	WideResNet-38	78.4
AC-Net	ResNet-101	82.3	DUC-HDC	ResNet-101	77.6
DeeplabV3	Xception-71	82.1	GCN	ResNet-101	76.9
DANet	ResNet-101	81.5	FoveaNet	ResNet-101	74.1
CCNet	ResNet-101	81.4	Ladder DenseNet	Ladder DenseNet-169	73.7
DeeplabV3	ResNet-101	81.3	RefineNet	ResNet-101	73.6
SPGNet	2xResNet-50	81.1	DeeplabV2	ResNet-101	70.4

2.4 Розробка моделі АСК автомобілем

Система автоматичного керування автомобілем складається з багатьох складних модулів кожен з яких потрібно згрупувати у єдину систему та розбити на прості блоки перед тим як перейти до реалізації. Архітектура системи буде використовувати інформацію, надану апаратними компонентами, і дозволить досягти мети - автономне керування. Загалом систему можливо поділити на наступні п’ять програмних модулів: сприйняття навколишнього середовища, картографування навколишнього середовища, планування руху, управління транспортним засобом і, нарешті, системний керівник. Загальна схема системи показана на рисунку 2.12.

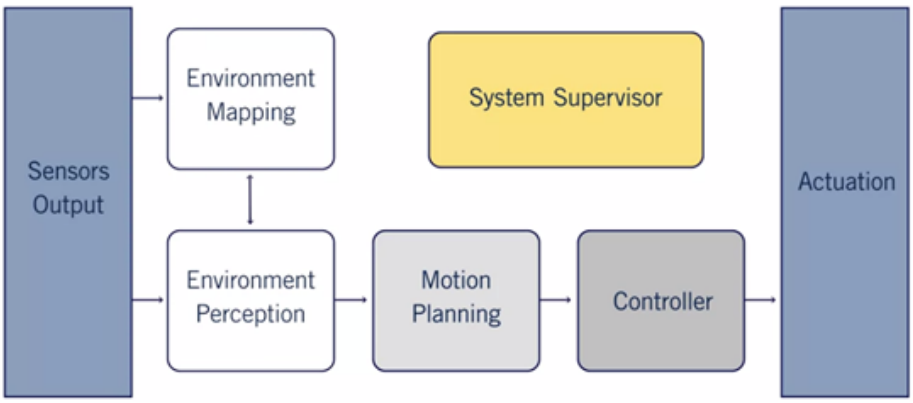

Рисунок 2.12 - Загальна схема системи

Це не єдина архітектура програмного забезпечення, що використовується в автономному русі, але є гарним поданням необхідних функцій для повної автономності автомобіля. Якщо подивитися на схему більш детально, то на вході автомобіль спостерігає за навколишнім середовищем, використовуючи різноманітні датчики. Вихідні вимірювання датчиків передаються у два набори модулів, призначених для розуміння довкілля навколо автомобіля. Модулі сприйняття навколишнього середовища мають два ключові обов'язки, по-перше, ідентифікація поточного розташування автономного транспортного засобу в космосі, і по-друге, класифікація та визначення важливих елементів навколишнього середовища для водіння. Прикладами таких елементів можливо вважати інші автомобілі, велосипеди, пішоходів, дорогу, дорожню розмітку та дорожні знаки, все, що безпосередньо впливає на рух. Модуль картографування навколишнього середовища створює набір карт, які знаходять об'єкти в середовищі навколо автономного транспортного засобу для цілого ряду різних цілей, від уникнення зіткнень до відстеження рухів та планування руху. Третій модуль відомий як планування руху. Модуль планування руху приймає всі рішення щодо того, які дії робити і куди рухатись, виходячи з усієї інформації, що надається модулями сприйняття та картографування. Основним результатом роботи модуля

планування руху є безпечний, ефективний та зручний спланований шлях, який рухає транспортний засіб до своєї мети. Потім запланований шлях виконується четвертим модулем - контролером. Модуль контролера бере шлях і визначає найкращий кут повороту, положення дросельної заслінки, положення педалі гальма та налаштування передач, щоб точно слідувати запланованому шляху. П’ятим і останнім модулем є наглядач системи - супервайзер. Супервайзер системи контролює всі частини стеку програмного забезпечення, а також апаратні результати, щоб переконатися, що всі системи працюють за призначенням. Також супервайзер системи відповідає за інформування водія про будь-які проблеми, виявлені в системі. Тепер, коли є загальне уявлення про функції основних модулів, потрібно детальніше розглянути кожен модуль окремо.

Спочатку потрібно почати з модуля сприйняття навколишнього середовища. Як вже обговорювалося раніше, є дві важливі частини стека сприйняття: локалізація транспортного засобу у просторі, класифікація та визначення важливих елементів навколишнього середовища. Модуль локалізації приймає безліч потоків інформації, таких як поточне розташування GPS, вимірювання IMU та одометрію колеса. Потім вони поєднуються, щоб отримати точне місце розташування автомобіля. Для більшої точності деякі модулі локалізації також включають дані LIDAR та камери. Як правило, проблема класифікації та локалізації елементів середовища ділиться на два сегменти, по-перше, виявлення динамічних об'єктів у навколишньому середовищі, а по-друге, виявлення статичних об'єктів у навколишньому середовищі. Модуль динамічного розпізнавання об’єктів використовує набір входів камери, а також хмари точок LIDAR для створення 3D-обмежувальних рам навколо динамічних об’єктів на сцені. Обмежувальні рамки 3D кодують клас або тип об’єкта, а також точне положення, орієнтацію та розмір об’єкта. Після виявлення динамічні об’єкти з часом відстежуються модулем відстеження. Модуль трекера надає не тільки поточне положення динамічних об'єктів, а й історію їх

шляху через навколишнє середовище. Історія шляху використовується разом із дорожніми картами для того, щоб передбачити майбутній шлях усіх динамічних об’єктів. Зазвичай цим займається модуль прогнозування, який поєднує всю інформацію щодо динамічного об’єкта та поточного середовища для прогнозування шляху всіх динамічних об’єктів. Модуль виявлення статичних об’єктів також покладається на комбінацію вхідних даних камери та даних LIDAR для ідентифікації значущих статичних об’єктів на сцені. Важливими даними можливо вважати поточну смугу руху, в якій знаходиться самохідний транспортний засіб, та розташування регулюючих елементів, як знаки та світлофори. Таким чином детальну схему модуля сприйняття навколишнього середовища можливо побачити на рисунку 2.13.

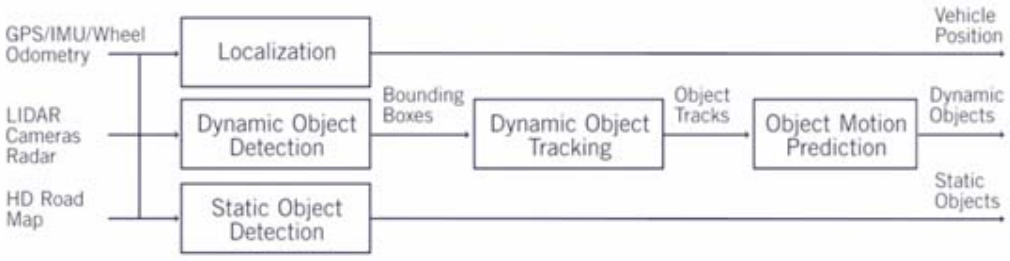

Рисунок 2.13 - Детальна схема модуля сприйняття навколишнього середовища

Тепер, коли краще зрозуміло, як працює стек сприйняття, необхідно перейти до модулів відображення середовища. Карти навколишнього середовища створюють кілька різних типів представлення поточного середовища навколо автономного автомобіля. Існує три типи карт: карта заповнення, карта локалізації та детальна дорожня карта. Сітка заповнення - це карта всіх статичних об’єктів у навколишньому транспортному середовищі. LIDAR переважно використовується для побудови сіткової карти заповнення. Набір фільтрів спочатку застосовується до даних LIDAR, щоб зробити їх придатними для використання в сітці заповнення.

Карта сітки заповнення представляє середовище як набір кліток сітки та асоціює ймовірність того, що кожна клітка зайнята. Це дозволяє обробляти невизначеність даних вимірювань та покращувати карту з часом. Карта локалізації, яка побудована з LIDAR, або даних камери, використовується модулем локалізації для покращення оцінки екологічного стану. Дані датчиків порівнюються з цією картою під час руху, щоб визначити рух автомобіля щодо карти локалізації. Потім цей рух поєднується з іншою інформацією з датчиків для точної локалізації автомобіля. Детальна дорожня карта містить карту відрізків доріг, які відображають середовище руху, через яке в даний час рухається автономний транспортний засіб. Дорожня карта фіксує знаки та розмітку смуги таким чином, що може бути використана для планування руху. Ця карта традиційно є поєднанням попередньо записаних даних, а також вхідної інформації з поточного статичного середовища, зібраної стеком сприйняття. Модулі картографування середовища та сприйняття суттєво взаємодіють для покращення роботи обох модулів. Наприклад, модуль сприйняття надає інформацію про статичне середовище, необхідну для оновлення детальної дорожньої карти, яка потім використовується модулем прогнозування для створення більш точних динамічних прогнозів об'єктів. Таким чином детальну схему модуля картографування навколишнього середовища можливо побачити на рисунку 2.14.

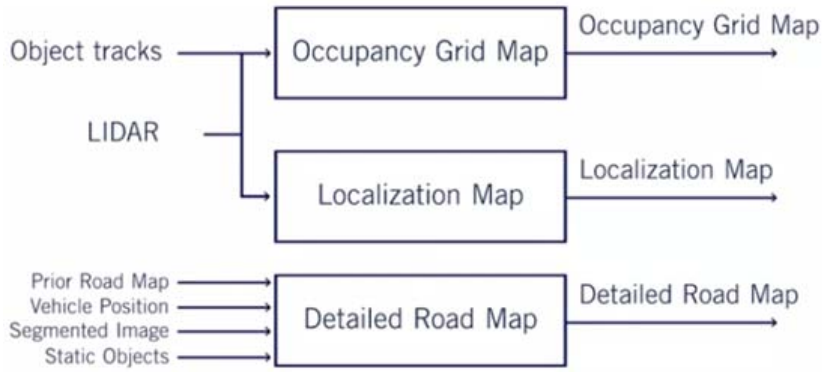

Рисунок 2.14 - Картографування навколишнього середовища

Тепер потрібно детальніше розглянути, як вихідні дані як з карт оточення, так і з модулів сприйняття поєднуються та використовуються модулем планування руху для створення шляху через середовище. Планування руху для самокерованих автомобілів є складним завданням, і його важко вирішити за один інтегрований процес. Натомість у більшості самокерованих автомобілів сьогодні використовується декомпозиція, яка розділяє проблему на кілька шарів абстракції наступним чином. На найвищому рівні планувальник місій займається довгостроковим плануванням та визначає місію по всьому горизонту водіння, починаючи від поточного місця розташування, через дорожню мережу до кінцевого пункту призначення. Щоб знайти повну місію, планувальник місії визначає оптимальну послідовність відрізків доріг, що з'єднують початок та пункт призначення, а потім передає це наступному шару. Планувальник поведінки - це наступний рівень абстракції, який вирішує короткострокові проблеми планування. Планувальник поведінки відповідає за встановлення набору безпечних дій або маневрів, які слід виконати під час подорожі по шляху місії. Прикладом рішень планувальника поведінки може бути те, чи транспортний засіб повинен змінити смугу, враховуючи бажану швидкість та прогнозовану поведінку сусідніх транспортних засобів. Поряд з маневром прийняття рішень планувальник поведінки також надає набір обмежень, які слід виконувати з кожною дією, наприклад, як довго залишатися на поточній смузі перед переходом. Нарешті, місцевий планувальник виконує негайне або реактивне планування і відповідає за визначення конкретного шляху та профілю швидкості руху. Подорож повинна бути гладкою, безпечною та ефективною, а також враховувати всі існуючі обмеження, що накладаються навколишнім середовищем та маневром. Для того, щоб створити такий план, місцевий планувальник поєднує інформацію, надану планувальником поведінки, сіткою заповнення, границями експлуатації транспортного засобу та іншими динамічними об’єктами в навколишньому середовищі. Результатом роботи

місцевого планувальника є запланована траєкторія, яка являє собою комбінований бажаний шлях та профіль швидкості на короткий проміжок часу в майбутньому. Таким чином детальну схему модуля планування руху можливо побачити на рисунку 2.15.

Рисунок 2.15 - Детальна схема модуля планування руху

Тепер потрібно подивитися, як типовий контролер транспортного засобу приймає задану траєкторію і перетворює її на набір точних команд спрацьовування для застосування транспортного засобу. Типовий контролер розділяє проблему управління на поздовжнє управління та бічне управління. Бічний регулятор видає кут повороту, необхідний для підтримання запланованої траєкторії руху, тоді як поздовжній регулятор регулює дросель, передачі та гальмівну систему для досягнення правильної швидкості. Обидва контролери обчислюють поточні помилки та ефективність відстеження локального плану та регулюють поточні команди спрацьовування, щоб мінімізувати помилки в майбутньому. Таким чином детальну схему модуля контролеру можливо побачити на рисунку 2.16.

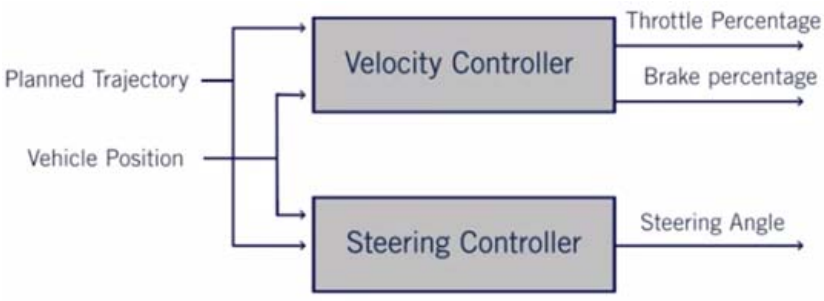

Рисунок 2.16 - Детальна схема модуля контролеру

Останній модуль системи супервайзер - це модуль, який постійно контролює всі аспекти автономного автомобіля та видає відповідне попередження у разі відмови підсистеми. Він ділиться на дві частини - апаратний нагляд і наглядач програмного забезпечення. Супервайзер апаратного забезпечення постійно контролює всі апаратні компоненти, щоб перевірити наявність будь-яких несправностей, таких як поламаний датчик, відсутність вимірювань або погіршення інформації. Інший обов'язок супервайзеру апаратного забезпечення полягає в постійному аналізі апаратних виходів, які не відповідають домену, в якому запрограмований автономний автомобіль. Наприклад, якщо один із датчиків камери заблокований паперовим пакетом або якщо сніг падає та пошкоджує дані хмари точок LIDAR. Супервайзер програмного забезпечення несе відповідальність за перевірку цього програмного пакета, щоб переконатися, що всі елементи працюють належним чином на належних частотах і забезпечують повні результати. Супервайзер програмного забезпечення також відповідає за аналіз невідповідностей між результатами роботи всіх модулів. Таким чином схему супервайзера у можливо побачити на рисунку 2.17.

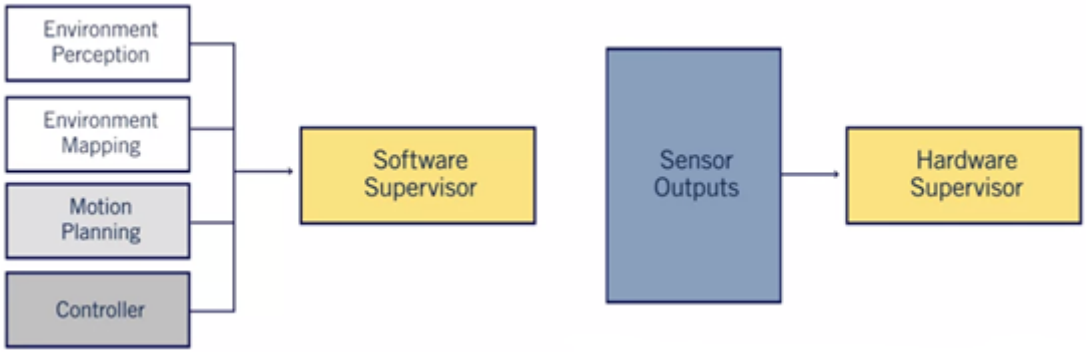

Рисунок 2.17 - Схема роботи супервайзера

Оскільки в даній дипломній роботі розробка та тестування АСК автомобілем робиться в симуляторі CARLA блок супервайзера не був реалізований.

2.5 Математичні моделі підсистем і елементів системи управління автомобілем

2.5.1 Поздовжня модель транспортного засобу

Модель приймає входи дросельної заслінки і крокує через поздовжні динамічні рівняння. Вхідні дані для моделі складають відсоток дросельної заслінки $x_\theta \in [0,1]$, який забезпечує обертаючий момент двигуна і згодом прискорює транспортний засіб для руху вперед. Динамічні рівняння складаються з багатьох етапів для перетворення входів дросельної заслінки на частоту обертання колеса: двигун, перетворювач крутного моменту, трансмісія, колесо. Ці каскади об'єднані в один інерційний термін J_e, який буде використовуватися в наступних комбінованих динамічних рівняннях двигуна:

$$\begin{aligned} J_e\dot{\omega}_e &= T_e - (GR)(r_{eff}F_{load}) \\ m\ddot{x} &= F_x - F_{load} \end{aligned} \tag{2.10}$$

де T_e - обертаючий момент двигуна;

GR - передавальне число;

r_{eff} - ефективний радіус;

m - маса транспортного засобу;

x - положення транспортного засобу;

F_x - сила шини;

F_{load} - загальна сила навантаження.

Обертаючий момент двигуна обчислюється на основі вхідного дроселя та кутової швидкості двигуна ω_e за допомогою спрощеної квадратичної моделі:

$$T_e = x_\theta(a_0 + a_1\omega_e + a_2\omega_e^2) \tag{2.11}$$

Сила навантаження складається з аеродинамічного опору F_{aero}, тертя R_x та сили тяжіння F_g під кутом α. Аеродинамічний опір - квадратична модель, а тертя - лінійна модель:

$$\begin{aligned} F_{load} &= F_{aero} + R_x + F_g \\ F_{aero} &= \frac{1}{2}C_a\rho A\dot{x}^2 = c_a\dot{x}^2 \\ R_x &= N(\hat{c}_{r,0} + \hat{c}_{r,1}|\dot{x}| + \hat{c}_{r,2}\dot{x}^2) \approx c_{r,1}\dot{x} \\ F_g &= mg\sin\alpha \end{aligned} \tag{2.12}$$

Абсолютне значення ігнорується для тертя, оскільки модель використовується лише для руху вперед. Зусилля шини обчислюються за допомогою рівнянь швидкості обертання двигуна та ковзання колеса:

$$\begin{aligned} \omega_w &= (GR)\omega_e \\ s &= \frac{\omega_w r_e - \dot{x}}{\dot{x}} \\ F_x &= \left\{ \begin{array}{ll} cs, & |s| < 1 \\ F_{max}, & \text{otherwise} \end{array} \right\} \end{aligned} \tag{2.13}$$

де ω_w - кутова швидкість руху колеса;

s - коефіцієнт ковзання.

Для комбінованих динамічних рівнянь двигуна разом із рівняннями сили в комірці за один крок функція приймає дросельну заслінку x_θ та кут нахилу α як вхідні дані та виконує чисельне інтегрування для оновлення змінних стану. Швидкість зближується до фіксованого значення на основі введення дросельної заслінки через аеродинамічний опір та обмеження сили шини. Подібний профіль швидкості можна побачити, встановивши від'ємний кут нахилу α. У цьому випадку гравітація прискорює транспортний засіб до кінцевої швидкості, де його врівноважує сила опору.

2.5.2 Модель рекурсивної оцінки положення транспортного засобу

Модель руху транспортного засобу отримує лінійну та кутову швидкості показань одометрії як вхідні дані та видає стан (тобто 2D-позицію) транспортного засобу:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + T \begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} + \mathbf{w}_k \right), \quad \mathbf{w}_k = \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2.14}$$

де x_k - поточна 2D-поза автомобіля;

v_k та ω_k - показання одометрії лінійної та кутової швидкості, які використовуються як вхідні дані до моделі руху.

Шум w_k має нормальний розподіл з постійною коваріацією Q. Модель вимірювання пов'язує поточну позу автомобіля з діапазоном LIDAR та вимірами підшипників:

$$\mathbf{y}_k^l = \begin{bmatrix} \sqrt{(x_l - x_k - d\cos\theta_k)^2 + (y_l - y_k - d\sin\theta_k)^2} \\ atan2\left(y_l - y_k - d\sin\theta_k, x_l - x_k - d\cos\theta_k\right) - \theta_k \end{bmatrix} + \mathbf{n}_k^l, \quad \mathbf{n}_k^l = \mathcal{N}(\mathbf{0}, \mathbf{R}) \quad (2.15)$$

де x_l та y_l - основні координати орієнтира l;

x_k, y_k і θ_k - представляють поточну позу транспортного засобу;

d - відома відстань між центром та LIDAR.

Шум n^l_k має нормальний розподіл з постійною коваріацією R.

Далі модель ділиться на: крок корекції та крок прогнозування. Для кроку корекції реалізується функція оновлення вимірювань, яка бере доступне вимірювання орієнтира l та оновлює поточну оцінку стану $\check{x}_k$. Для кожного вимірювання орієнтиру, отриманого за певний крок часу k, були виконані наступні кроки:

- Обчислення моделі вимірювання при $\check{x}_k$:

$$\begin{gathered} \mathbf{y}_k^l = \mathbf{h}(\mathbf{x}_k, \mathbf{n}_k^l) \\ \mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}_k}\Big|_{\check{\mathbf{x}}_{k,0}}, \quad \mathbf{M}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{n}_k}\Big|_{\check{\mathbf{x}}_{k,0}} \end{gathered} \quad (2.16)$$

- Обчислення приросту Калмана:

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{H}_k^T \left(\mathbf{H}_k \check{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^T\right)^{-1} \quad (2.17)$$

- Виправлення прогнозованого стану:

$$\begin{aligned}\check{\mathbf{y}}_k^l &= \mathbf{h}\left(\check{\mathbf{x}}_k, \mathbf{0}\right) \\ \hat{\mathbf{x}}_k &= \check{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{y}_k^l - \check{\mathbf{y}}_k^l\right)\end{aligned} \tag{2.18}$$

- Виправлення коваріації:

$$\hat{\mathbf{P}}_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \check{\mathbf{P}}_k \tag{2.19}$$

Для кроку прогнозування використовується основний цикл фільтра, визначаючий крок прогнозування EKF використовуючи надану модель руху:

$$\begin{aligned}\check{\mathbf{x}}_k &= \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right) \\ \check{\mathbf{P}}_k &= \mathbf{F}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T .\end{aligned} \tag{2.20}$$

$$\mathbf{F}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0}, \quad \mathbf{L}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}_k}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0}$$

2.5.3 Модель для вимірювання стерео глибини до сценарію водіння

Щоб оцінити глибину стерео сцени, треба використовувати таку послідовність дій:

- визначити диспропорцію між двома зображеннями;

- розкласти матриці проекції на внутрішню матрицю камери K і зовнішню інформацію R, t;

- оцінити глибину, використовуючи вхідні дані.

Визначення диспропорції між двома зображеннями буде робитися за допомогою зовнішніх бібліотек на мові Python, тому в даній дипломній роботі детальний опис цього блоку опускається.

Розкладання проекційної матриці P можливо наступним чином: якщо припустити P як комбінацію внутрішніх параметрів K, зовнішнього обертання R і перекладу t тоді:

$$P = K[R|t] \tag{2.21}$$

Обернена матриця до KR дозволяє здійснити QR-декомпозицію, щоб отримати R^{-1} і K^{-1}:

$$(KR)^{-1} = R^{-1}K^{-1} \tag{2.22}$$

Звідси здавалося б, ніби можливо легко визначити K, R і t. На жаль, це не так просто, бо QR-декомпозиція не є унікальною. Це призводить до того, що доводиться перевіряти знаки діагоналі матриці K і правильно регулювати R. Також повинно робити твердження щодо напрямків камери та зображень осей x, y та z.

Після множення матриці однорідний компонент w_c, як правило, не буде дорівнює 1. Отже, для відображення назад у дійсну площину повинно виконувати однорідне ділення або ділення в перспективі шляхом ділення кожного компонента на w_c.

Для того, щоб створити карту глибини з пари зображень, зроблених за допомогою стерео камери також потрібно:

- отримати фокусну відстань f з матриці K;
- обчислити базову лінію b, використовуючи відповідні значення з векторів перекладу t;
- обчислити карту глибини зображення за наступною формулою:

$$Z = \frac{fb}{x_L - x_R} = \frac{fb}{d} \tag{2.23}$$

де d - карта диспропорцій.

2.5.4. Модель оцінки керованого простору з використанням результатів семантичної сегментації

Для оцінки керованого простору з використанням результатів семантичної сегментації в 3D вихідними даними були семантичні сегментації, матриця калібрування камери K, а також глибина на піксель. Розрахунок координат x, y та z кожного пікселя на зображенні виконувався за формулою:

$$\begin{aligned} z &= depth \\ x &= \frac{(u-c_u)*z}{f} \\ y &= \frac{(v-c_v)*z}{f} \end{aligned} \tag{2.24}$$

де c_u - внутрішні параметри калібрування;

c_v - внутрішні параметри калібрування;

f - внутрішні параметри калібрування.

Ці внутрішні параметри калібрування, знайдені в матриці калібрування камери K такі, що:

$$K = \begin{pmatrix} f & 0 & u_c \\ 0 & f & u_v \\ 0 & 0 & 1 \end{pmatrix} \tag{2.25}$$

3 ПРАКТИЧНА ЧАСТИНА

Оскільки на мові програмування Python є багато вбудованих інструментів та бібліотек для роботи з нейронними мережами, даний язик був вибраний для написання дипломної роботи в середовищі розробки PyCharm, котра має додаткові інструменти для комфортної розробки та тестування написаного коду.

3.1 Постановка задачі реалізації

Реалізація автопілоту - великий проект. Але на базі отриманого досвіду при написані бакалаврського диплому на цю тему та отриманого нового досвіду був розроблений наступний план:

- розробити передню поздовжню автомобільну модель. Модель приймає на вході дросельну заслінку і крокує через поздовжні динамічні рівняння. Після реалізації моделі, протестувати на наборі входів, які містять дані про рух по невеликому схилу;

- написати та впровадити до системи контролер для симулятора CARLA. Мета даного контролера - керувати транспортним засобом, щоб він слідував за гоночною трасою, переміщаючись по заданих пунктах. Автомобіль повинен досягти цих точок на певних бажаних швидкостях, тому буде потрібно як поздовжній, так і поперечний контроль. Разом з іншими змінними, шляхові точки будуть оновлюватимуться на кожному етапі моделювання - тому, при написані контролера потрібно враховувати, що змінна маршрутної точки постійно змінюється. Маршрутні точки - це лінійно інтерпольована (для розташування та швидкості) підмножина усього набору шляхових точок. Іншими словами, змінна маршрутних точок - це покращена (з більш чіткою роздільною здатністю) частина всього набору шляхових точок, що знаходиться поблизу автомобіля. Це робиться для зменшення часу обчислення та продуктивності контролера.

Бажана швидкість обчислюється як швидкість маршрутної точки, що знаходиться найближче до транспортної точки;

- використовуючи цю інформацію для контролера, настроїти роботу з дроселем, кермом та гальмом автомобіля. Обробити всі вимірювання від CARLA щодо центрального положення автомобіля;

- після реалізації, підключитися до сервера, запустити контролер. Проробити зворотний зв'язок та траєкторію. Зворотній зв'язок траєкторії буде містити автомобіль, початкове і кінцеве положення, весь пройдений шлях та невелику затінену область, яка позначає підмножину інтерпольованих точок, які слід надіслати в контролер для оновлення управління. Лінійна інтерполяція використовується між маршрутними точками, щоб забезпечити більш точний запит щодо дозволенної швидкості для контролера. Зворотний зв'язок управління відображає дросельну заслінку, рульове управління та гальмові виходи, а також реакцію швидкості для моделювання. Це загальний відгук для перегляду того, що клієнт надсилає на сервер CARLA з точки зору команд управління. Бажану швидкість встановити на найближчу інтерпольовану точку швидкості до поточного положення автомобіля;

- реалізувати оцінку навколишнього середовища та локалізацію для автопілоту. Для цього реалізувати метод найменших квадратів та розробити рекурсивний його аналог. Для оцінки траєкторії руху транспортного засобу застосувати розширений фільтр Калмана, який оцінить траєкторію руху автомобіля, використовуючи вимірювання одометрії, дальності та підшипників;

- для оцінки стану транспортного засобу на проїжджій частині застосувати розширений фільтр Калмана за станом помилки (ES-EKF), щоб локалізувати транспортний засіб, використовуючи дані симулятора CARLA. Даний етап проекту складається з трьох частин:

1) Реалізувати ES-EKF, фільтр для корекції кроку. Фільтр покладається на дані IMU для розповсюдження стану в часі, а також на

оновлення положення GPS і LIDAR для корекції оцінки стану. Візуалізувати результати оцінки та порівняти їх із положенням транспортного засобу. Повну реалізацію фільтра перевірити шляхом порівняння передбачуваного положення транспортного засобу для частини траєкторії.

2) Вивчити вплив помилки калібрування датчика на оцінки пози автомобіля. Зокрема, навмисно змінити трансформацію між кадром датчика LIDAR та кадром датчика IMU; використання неправильного перетворення призведе до помилок в оцінках положення автомобіля. Після розгляду помилок, визначити, як відрегулювали параметри фільтра (дисперсії шуму), щоб спробувати зменшити ці помилки.

3) Дослідити наслідки відмови датчика, тобто коли вся інформація про зовнішнє позиціонування (від GPS та LIDAR) втрачається протягом короткого періоду часу. Мета цієї частини - проаналізувати, як втрата зовнішніх поправок призводить до дрейфу в оцінці положення автомобіля, а також допоможе зрозуміти, як змінюється невизначеність оцінки положення, коли вимірювання датчика недоступні.

- розробити візуальну одометрію для локалізації автопілоту. Це необхідно для оцінки автономної траєкторії транспортного засобу за зображеннями, зробленими за допомогою монокулярної камери, встановленої на транспортному засобі;

- розробити окупаційну сіткову генерацію. Це блок картування для планування. Необхідно створити сітку заповнення за допомогою вимірювань сканера лідара з транспортного засобу, що рухається в невідомому середовищі. Використати модель вимірювання зворотного сканера, розроблену раніше, щоб відобразити ці вимірювання на ймовірність заповнення, а потім виконати ітеративні оновлення логістів на карті переконань у сітці заповнення. Після того, як автомобіль назбирає достатньо даних, сітка заповнюваності повинна сходитися з справжньою картою;

- розробити алгоритм пошуку найкоротший шляхів дорожньої мережі. Спробувати алгоритм пошуку Дейкстри. Потім змінити цей алгоритм, використовуючи евристику відстані, щоб виконати пошук A*. Отриманий найкоротший шлях порівняти із розробленим в ручну шляхом;

- дореалізовувати концепції поведінки та місцеве планування. Завершити функціональний стек планування руху, який буде дозволяти уникнути як статичних, так і динамічних перешкод при відстеженні центральної лінії смуги , при цьому також обробляючи знаки зупинки. Для цього доведеться застосувати логіку планування поведінки, а також статичну перевірку зіткнень, вибору шляху та генерацію профілю швидкості;

- останнім етапом необхідно проаналізувати результати роботи автопілоту та зробити висновки.

3.2 Реалізація поставленої задачі

Першою в списку є реалізація поздовжньої моделі транспортного засобу. Автомобіль починає рух з початковою швидкістю 5 м/с і частотою обертання двигуна 100 рад/с, для чисельної інтеграції використовується час вибірки 10 мс. Потім було впроваджено комбіновані динамічні рівняння двигуна разом із рівняннями сили в комірці. Після реалізації цієї моделі, можливо надсилати постійні вводи дросельної заслінки в транспортний засіб. Код реалізації можливо побачити в: (додаток А, приклад 3.1).

Наступним кроком була розроблена модель для рекурсивної оцінки положення транспортного засобу вздовж траєкторії, використовуючи доступні вимірювання та модель руху.

Датасет має дані з датчика LIDAR, котрий передає системі виміри дальності та підшипників, що відповідають окремим орієнтирам у навколишньому середовищі. В ході реалізації припускається, що глобальні

положення орієнтирів відомі заздалегідь, а також відома асоціація даних, тобто те, яке вимірювання належить до якого орієнтиру.

Оскільки дані моделі нелінійні, був використаний розширений фільтр Калмана (EKF) як оцінювач стану. Таким чином модель можливо поділити на наступні етапи: етап прогнозування, який використовує вимірювання одометрії та модель руху для отримання оцінки стану та коваріації на даному етапі часу, і крок корекції, який використовує вимірювання дальності та підшипників, надані LIDAR, для корекції пози та оцінки коваріації пози. Для кроку корекції була реалізована функція оновлення вимірювань. Для кроку прогнозування був реалізований основний цикл фільтра, визначаючий крок прогнозування EKF.

Таким чином були реалізовані модель руху та модель оцінки, код котрих: (додаток А, приклад 3.2).

Наступним кроком потрібно було реалізувати модель для вимірювання стерео глибини до сценарію водіння. Це допоможе знаходити відстань до зіткнення з перешкодою. Для цього в модель на вхід буде приходити пара стереозображень. Приклад пари стереозображень показанні на рисунку 3.1.

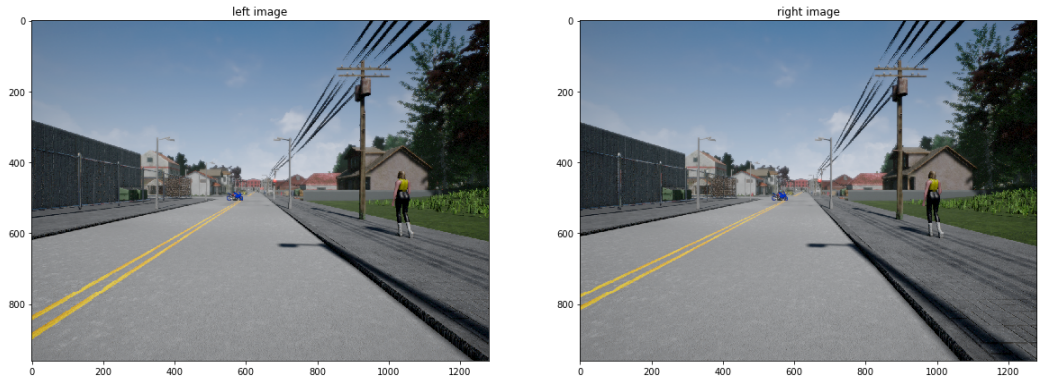

Рисунок 3.1 - Пара стереозображень з CARLA

При поєднанні пари стереозображень з рисунку 3.6 виходить результат показаний на рисунку 3.2.

Рисунок 3.2 – Поєднані стереозображення

Для визначення диспропорції між двома зображеннями на вхід подається пара стереозображень та повертається карта невідповідності з точки зору лівої камери. Для обчислення карти невідповідності використовувалась функція StereoSGBM з бібліотеки OpenCV. Для визначення диспропорції вхідні зображення заздалегідь перетворюють в сірий формат з RGB завдяки функції cvtColor з бібліотеки OpenCV.

Для того, щоб отримати уявлення про те, як параметри та вибір збігу змінюють результуючу карту невідповідності, була зроблена візуалізація показана на рисунку 3.3.

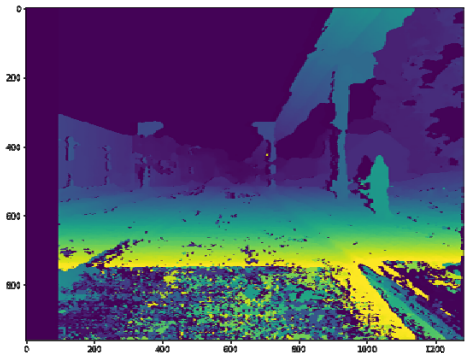

Рисунок 3.3 - Візуалізація глибини зображення

Наступна функція розкладає проекційні матриці. Завдяки наданої функції OpenCV - decomposeProjectionMatrix, функція перевіряє знаки

діагоналі матриці і правильно регулює зовнішню інформацію. Також вона робить твердження щодо напрямків камери та зображень осей x, y та z.

Хоча можливо мати карту глибини кожного пікселя на сцені, система ще не знає, які з цих пікселів безпечні (як дорога) чи потенційна перешкода (як мотоцикл). Щоб знайти ці об’єкти, був реалізований детектор об’єктів та класифікатор об’єктів. Код реалізації котрих можливо побачити в: (додаток А, приклад 3.3).

Також додатково для того, щоб система автоматично визначала, де знаходиться ця перешкода на місці події була використана перехресна кореляція. Алгоритм перехресної кореляції вимагає великих числових обчислень для кожного пікселя на зображенні, тому була використана вже оптимізована функція matchTemplate з OpenCV. За допомогою цієї теплової карти та використовуючи функцію minMaxLoc стає можливим вилучення положення перешкоди. Відображення результуючої теплової карти перехресної кореляції та координати розташування перешкоди зображені на рисунку 3.4.

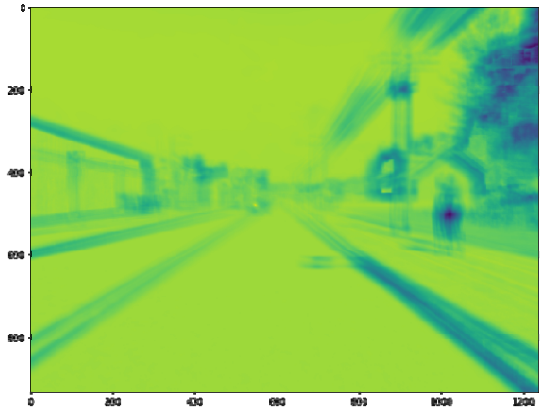

Рисунок 3.4 - Відображення результуючої теплової карти перехресної кореляції

Останнім кроком для реалізації цієї моделі була розроблена функція для обрізання ділянки карти глибини, що відповідає перешкоді, і знаходить

найближчі точки в цьому секторі. Код реалізації моделі розрахунку глибини: (додаток А, приклад 3.4)

Наступним кроком для системи потрібна візуальна одометрія для локалізації в автономному русі. Це необхідно, щоб оцінювати автономну траєкторію руху автомобіля за зображеннями, зробленими за допомогою монокулярної камери з датасету. Для цього потрібно:

- витягнути особливості з фотографій з датасету;
- використати витягнуті об’єкти, для того щоб знайти збіги між об’єктами на різних фотографіях;
- використати знайдені збіги, щоб оцінити рух камери між наступними фотографіями;
- використати приблизний рух камери для побудови траєкторії руху автомобіля.

Кожен кадр з датасету містить зображення RGB, карту глибини, зроблену з установкою на транспортному засобі та сіру версію зображення RGB, яка використовується для обчислення. Максимальна відстань до глибини - 1000. Це значення глибини показує, що вибраний піксель знаходиться на відстані щонайменше 1000 м (1 км) від камери, однак точна відстань цього пікселя від камери невідома. Наявність такого роду балів при подальшій оцінці траєкторії може вплинути на точність траєкторії. Використовуючи ці дані була реалізована функція для вилучення особливостей.

Наступним кроком після вилучення об’єктів на кожному зображенні є узгодження об’єктів з наступними кадрами. Для цього була впроваджена функція, котра аналізує відповідність особливостей для пари зображень. Тепер на даному етапі є все, щоб виконати візуальну одометрію для автономного автомобіля. Тому настав час для оцінки траєкторії, оцінюючи позу автомобіля, вивчаючи зміни, які викликає рух на зображеннях бортової камери. Для цього була реалізована функція, яка оцінює рух

камери з пари зображень. Код реалізації моделі розрахунку траєкторії: (додаток А, приклад 3.5)

Написані раніше моделі системи слугують добувачами матеріалу корисної інформації про сцену необхідні для сприйняття довкілля для самокерованих автомобілів, котрі дозволяють безпечно та надійно об'їжджати перешкоди з навколишнього середовища. На наступному етапі необхідно було реалізувати вихідні дані семантичної сегментації нейронних мереж для оцінки провідного простору в 3D, оцінки смуги, фільтрації помилок на виході 2D-детекторів об’єктів, визначення відстані перешкод від самохідного автомобіля. Для цього була реалізована функція оцінки площини землі за допомогою RANSAC. У контексті самокерованих автомобілів, проїзний простір включає будь-який простір, який автомобіль фізично здатний проїхати в 3D. Завдання оцінки рухомого простору еквівалентно оцінці пікселів, що належать площині землі в сцені. Тому був використаний RANSAC для оцінки площини землі в рамці координат 3D-камери з координат x, y та z, оцінених вище. Першим для цього кроком є обробка результату семантичної сегментації для вилучення відповідних пікселів, що належать до класу, який потрібно розглянути як основний. Для цієї оцінки цим класом називається клас дороги з індексом відображення 7. Наступним кроком є використання витягнутих координат x, y та z пікселів, що належать дорозі, для оцінки площини землі. RANSAC був використаний для стійкості проти викидів. Цей крок ділиться на 6 блоків:

- вибрати навмання мінімум 3 точки координат;
- обчислити модель площини землі, використовуючи вибрані випадкові точки;
- обчислити відстань від моделі наземної площини до кожної точки і обчислити кількість внутрішніх елементів на основі порогу відстані;

- перевірити, чи поточна кількість вставок не перевищує всіх попередніх ітерацій, і зберегти набір вставок із найбільшою кількістю точок;
- повторювати, поки кількість ітерацій ≥ заданій кількісті ітерацій або кількість вставок ≥ мінімальній кількісті вставок;
- повторно обчислити і повернути модель площини, використовуючи всі внутрішні елементи в остаточному наборі.

Також була зроблена функція для візуалізації передбачуваного простору для керування в 3D. Однак оцінки проїзного простору недостатньо для того, щоб самохідний автомобіль виїжджав на дороги. Самостійний автомобіль все ще повинен провести оцінку смуги руху, щоб знати, де йому дозволено законно їздити. Для цього була реалізована функція, котра дозволяє оцінювати смуги з використанням результату семантичної сегментації. Першим кроком для виконання цього була оцінка будь-якого рядка, який відповідає межі смуги, використовуючи результати семантичної сегментації. У подальшому ці рядки будуть називатися «пропозиціями». Цей крок ділився на 3 задачі:

- створити зображення, що містить семантичні пікселі сегментації, що належать до категорій, що мають відношення до меж смуги, подібно до того, що було зроблено раніше для площини дороги. Для цієї оцінки ці пікселі мають значення 6 та 8 на виході сегментації нейронної мережі;
- виконати виявлення краю на похідному зображенні межі смуги;
- виконати оцінку лінії на виході виявлення краю.

В ході тестування було виявлено, що модель виявила одну непотрібну смугу. Тому для того, щоб покращити результат роботи була також добавлена функція для злиття та фільтрації смуг ліній. Її роль для виконання оцінки поточної межі смуги полягає у об’єднанні зайвих ліній та фільтруванні будь-яких горизонтальних ліній, що видно на зображенні. Об'єднання надлишкових ліній можна вирішити за допомогою групування ліній з однаковим нахилом та перехопленням. Горизонтальні лінії можна

відфільтрувати через порогове значення нахилу. Дана фільтрація проходить в три кроки:

- отримати нахил і перехоплення кожного рядка;
- визначити лінії з нахилом, меншим за поріг горизонтального нахилу;
- об’єднати всі рядки в кластери, використовуючи середнє усереднення.

В останньому кроку для покращення цієї моделі на виході повинен бути один рядок на смугу. Для цього була реалізована екстраполяція смуг, які починаються на початку дороги та закінчуються в кінці дороги, та функція визначення розмітки смуги, що належить поточній смузі.

Далі потрібно використовувати вихід 2D-виявлення об’єкта, щоб визначити мінімальну відстань до удару об’єктами на сцені. Однак завдання ускладняється тим фактом, що надані 2D-детектори здійснюються з використанням 2D-об'єкта з високим викликом та низькою точністю. Тому для реалізації був використаний семантичний результат сегментації, котрий допоможе визначати, які обмежувальні поля є дійсними. Потім була обчислена мінімальна відстань до удару, використовуючи обмежувальні рамки та глибинне зображення. Як правило кадр має зайві рамки, для їх усунення була написана функція фільтрації.

Кінцевим пунктом для цієї моделі є функція оцінки, котра рахує мінімальну відстань до кожного обмежувального блоку у вхідних детекторах. Це було реалізовано завдяки розрахунку різниці від мінімальної відстані від пікселів у обмежувальному вікні до центру камери. Повний код реалізації моделі сприйняття довкілля системи можливо знайти: (додаток А, приклад 3.6)

Наступною частиною системи була розроблена модель картування для планування. Під час цієї оцінки створюється сітка заповнюваності за допомогою вимірювань сканера лідара з транспортного засобу, що рухається в невідомому середовищі. Була використана модель вимірювання зворотного сканера, щоб відобразити ці вимірювання на

ймовірність заповнення, а потім виконати ітеративні оновлення логів на карті впевненості у сітці занятості. Після того, як автомобіль збирає достатньо даних про навколишнє середовище, сітка заповнюваності повинна сходитися до справжньої карти.

В ході реалізації цієї моделі були пройдені наступні етапи:

- збір вимірювань дальності руху автомобіля, що рухається, за допомогою функції сканування лідара;
- витяг інформації про заповнення з вимірювань дальності за допомогою інверсної моделі сканера;
- оновлення логів щодо місць заповнення на основі вхідних вимірювань.

Модель зворотного сканера повертає матрицю виміряних значень ймовірності заповнення на основі моделі сканування лідара. Функція get_ranges фактично повертає значення діапазонів для даного положення автомобіля та підшипника сканера. Далі йде ініціалізація необхідних змінних для моделювання. Це включає початковий стан, а також набір контрольних дій для автомобіля. Також встановлюється швидкість обертання сканування лідара. Перешкоди на істинній карті представлені у вигляді 1 на істинній карті, а 0 означають вільний простір. Кожна клітка на карті переконань m ініціалізується до 0,5 як ймовірність занятості, і з цієї карти переконань йде обчислення сітки заповнення логів L. (Додаток А, приклад 3.7)

Наступна модель системи - планування місії. Для цього був впроваджений у систему алгоритм пошуку Дейкстри на дорожній мережі в Берклі, штат Каліфорнія, оскільки у відкритому доступі містилися детальні дані даного штату. Також був проведений порівняльний аналіз роботи алгоритму Дейкстри, з алгоритмом розрахунку евристичної відстані, що виконує пошук A*.

Алгоритм Дейкстри: щоб виконати пошук Дейкстри, потрібна черга з пріоритетами (або мінімальна купа), яка визначається як клас. Цей клас

доступний як стандартний словник, за винятком того, що він упорядковує ключі за їх значенням. У цьому словнику були використані вершини як ключі до черги з пріоритетнами, а їх відстань від початку як їх значення. Основним входом для пошуку є графік, який представляє собою мережу доріг. Повернене значення - це список кортежів, кожен з яких представляє вихідний край. Другим елементом кожного кортежу є вихідна вершина на іншому кінці ребра. В ході реалізації функції пошуку обов'язково зберігались оптимальні попередники кожної вершини у словнику попередників, щоб можливо було отримати оптимальний шлях, як тільки буде знайдений вузол цілі.

Алгоритм A*: оскільки використовувались справжні дані карти, вони були перетворені у формат, який можливо використовувати для обчислення відстані. З кожною точкою даних пов'язані широта та довгота, які були перетворені в (x, y, z) координати на землі (для простоти було прийнято вважати землю сферою радіусом 6371 км). Завдяки цьому стає доступним можливість взяти пряму відстань між двома точками як найближчу відстань між ними. На невеликих відстанях такий метод є точним. Реалізацію моделі планування маршруту можливо знайти: (додаток А, приклад 3.8), а реалізацію допоміжних функцій в: (додаток А, приклад 3.9).

3.3 Результати досліджень

Реалізація поздовжньої моделі транспортного засобу була перевірена на наборі входів, в яких прописаний рух через невеликий схил дороги. Для цього був вибраний схил показаний на рисунку 3.5.

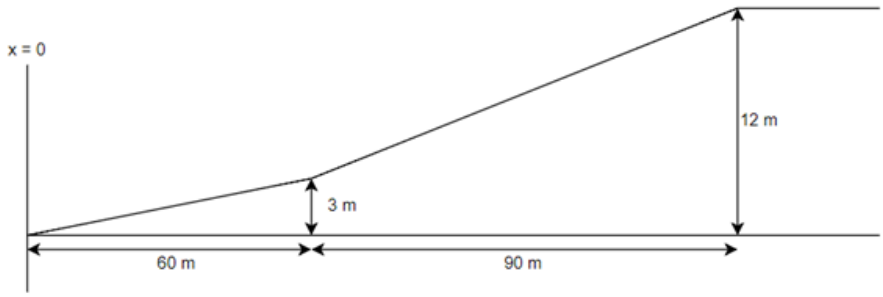

Рисунок 3.5 – Схема схилу

Для підйому по цьому схилу передбачений трапецієподібний вхід дроселя протягом наступних 20 секунд, як показано на рисунку 3.6.

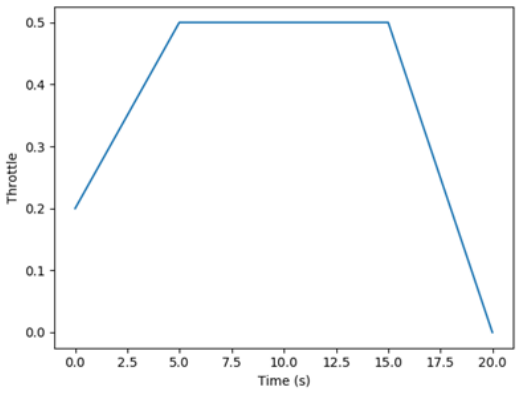

Рисунок 3.6 - Передбачений трапецієподібний вхід дроселя протягом 20 секунд

Автомобіль починає з 20% дросельною заслінкою і поступово збільшує до 50%. Так триває протягом 10 секунд, коли транспортний засіб піднімається на крутіший схил. Потім автомобіль зменшує дросель до 0. В результаті тестування було отримано наступні результати:

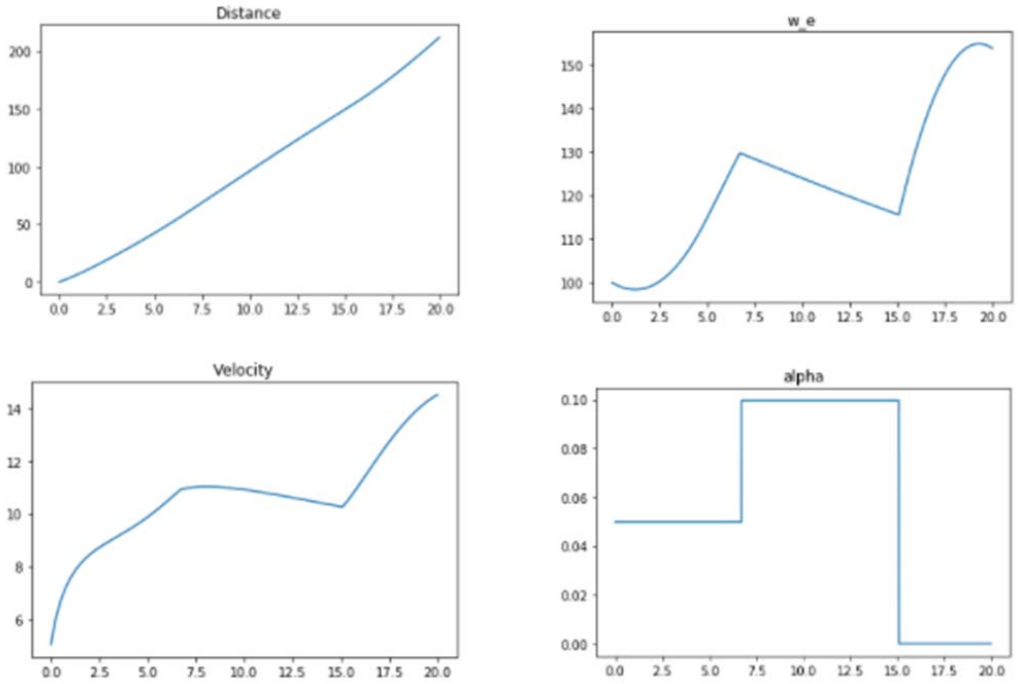

Рисунок 3.7 - Результати тестування

Проаналізувавши результати був зроблений висновок, що реалізація поздовжньої моделі транспортного засобу була зроблена правильно.

Результати роботи моделі руху та моделі оцінки можливо побачити на рисунку 3.8.

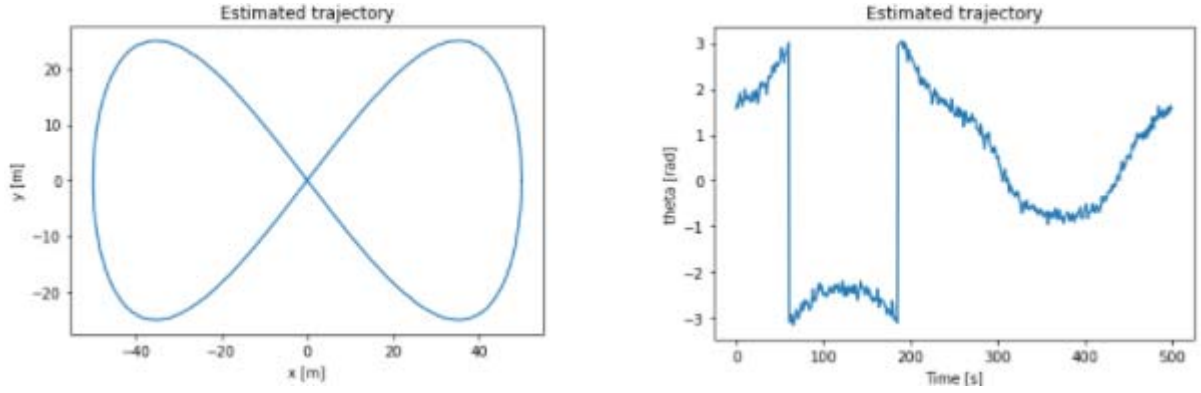

Рисунок 3.8 - Обчислення траєкторії

Результати роботи моделі побудови карти глибини можливо побачити на рисунку 3.9.

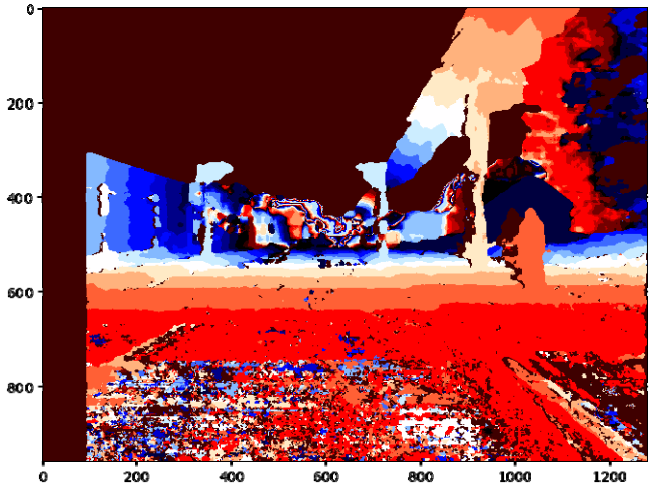

Рисунок 3.9 - Результат роботи карти глибини

Реалізація моделі сприйняття дорожніх об’єктів дає наступний результат візуалізації обмежувального поля показаний на рисунку 3.10.

Рисунок 3.10 - Візуалізація обмежувального поля

Результат роботи функції для вилучення особливостей показаний на рисунку 3.11.

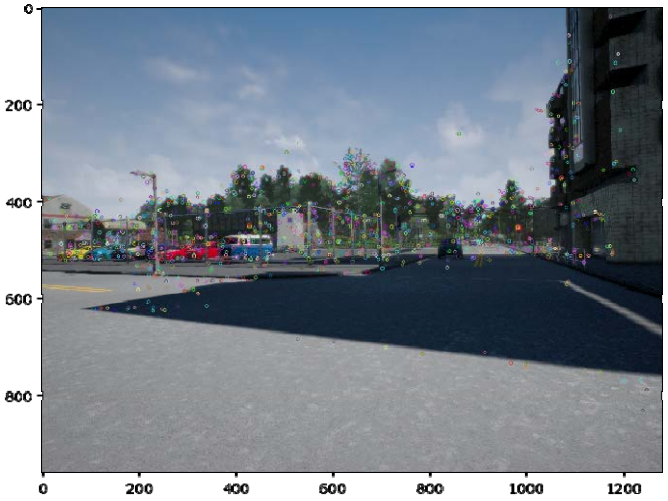

Рисунок 3.11 - Результат роботи функції для вилучення особливостей

Результат роботи функції знайдення збігів можливо побачити на рисунку 3.12.

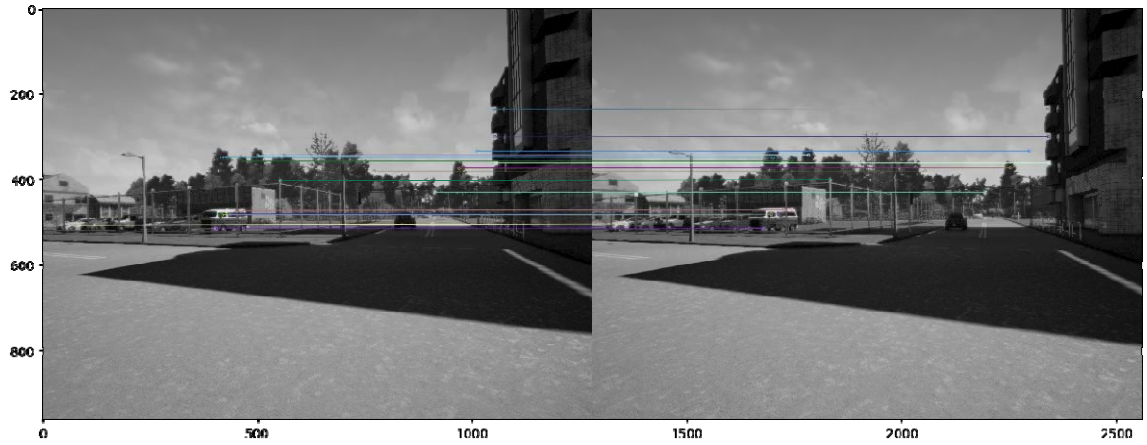

Рисунок 3.12 - Результат знайдених збігів

Результат зміни траєкторії можливо побачити на рисунку 3.13.

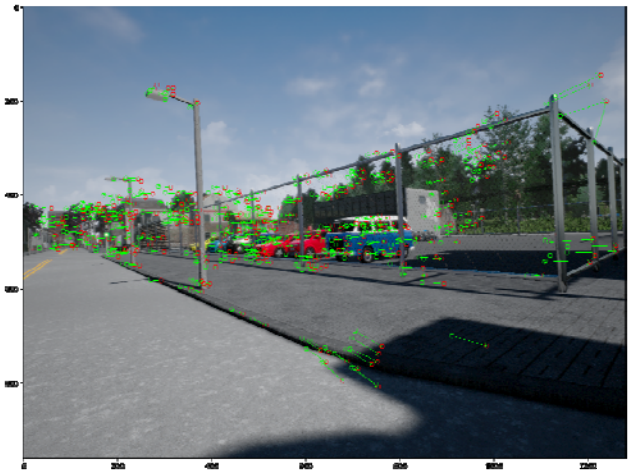

Рисунок 3.13 - Результат зміни траєкторії

Візуалізація траєкторії показана на рисунку 3.14.

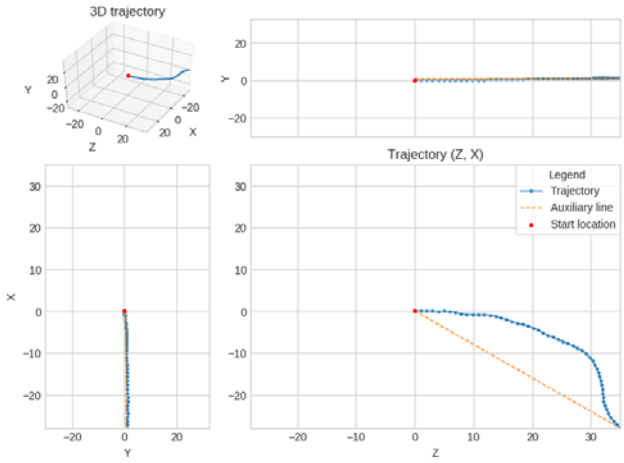

Рисунок 3.14 - Візуалізація траєкторії

Для того щоб перевірити правильність розрахункової площини, на рисунку 3.15 справа зображена візуалізація набору вставок, обчислених на всьому зображенні та результат отриманих координат дороги x, y, z показаний на рисунку 3.15 зліва.

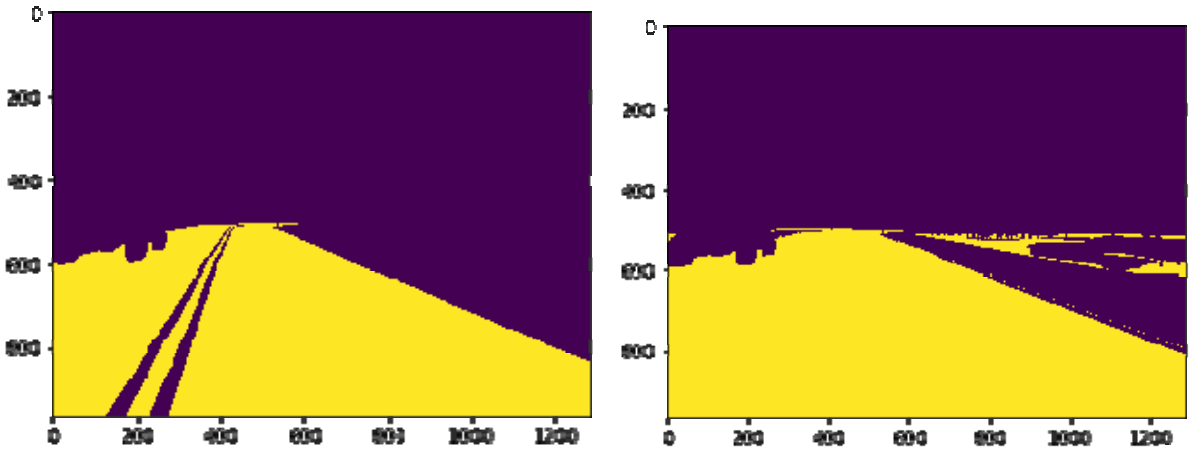

Рисунок 3.15 - Результат отриманих координат дороги x, y, z зліва та візуалізація методом набору вставок справа

Як можливо побачити з вище показаних рисунків модель розрахункової площини працює правильно.

Результат роботи функції візуалізації передбачуваного простору для керування в 3D показаний на рисунку 3.16.

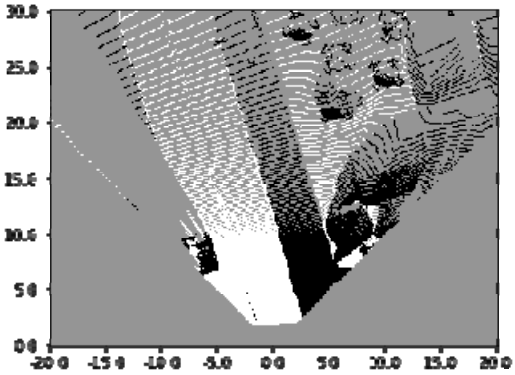

Рисунок 3.16 - Візуалізація передбачуваного простору для керування в 3D

Наведена вище візуалізація лише показує, куди фізично може їхати самохідний автомобіль.

Результат виявлення смуг можливо побачити на рисунку 3.17.

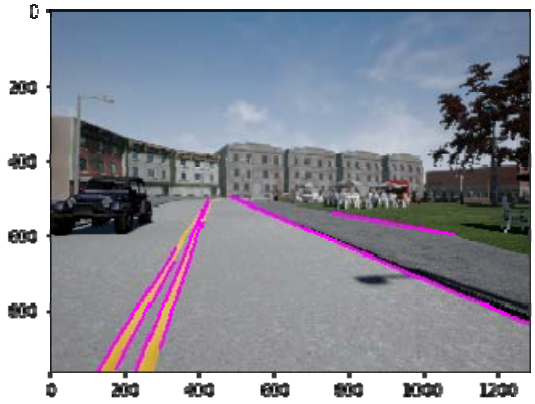

Рисунок 3.17 - Результат виявлення смуг на зображені

Як видно на рисунку 3.17 модель виявила одну непотрібну смугу, тому дана модель була удосконалена. Тому була добавлена фільтрація та екстраполяція смуг. Результат удосконаленої моделі виявлення смуг на дорозі можливо побачити на рисунку 3.18 зліва, а результат екстраполяції смуг можливо побачити на рисунку 3.18 справа.

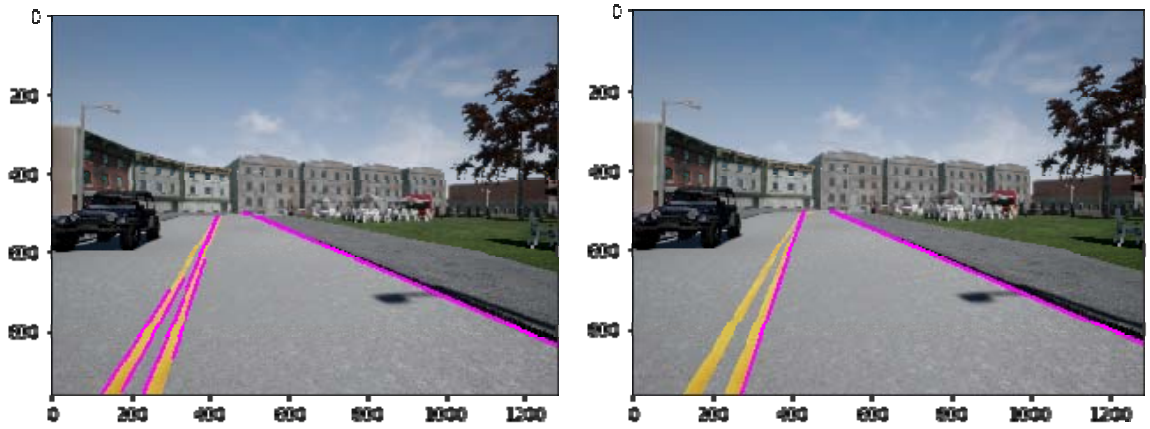

Рисунок 3.18 - Результат удосконаленої моделі виявлення смуг на дорозі зліва та результат екстраполяції смуг справа

Результат візуалізації виявлення вихідних даних для поточного кадру можливо побачити на рисунку 3.19 зліва, а результат роботи функції фільтрації можливо побачити на рисунку 3.19 справа.

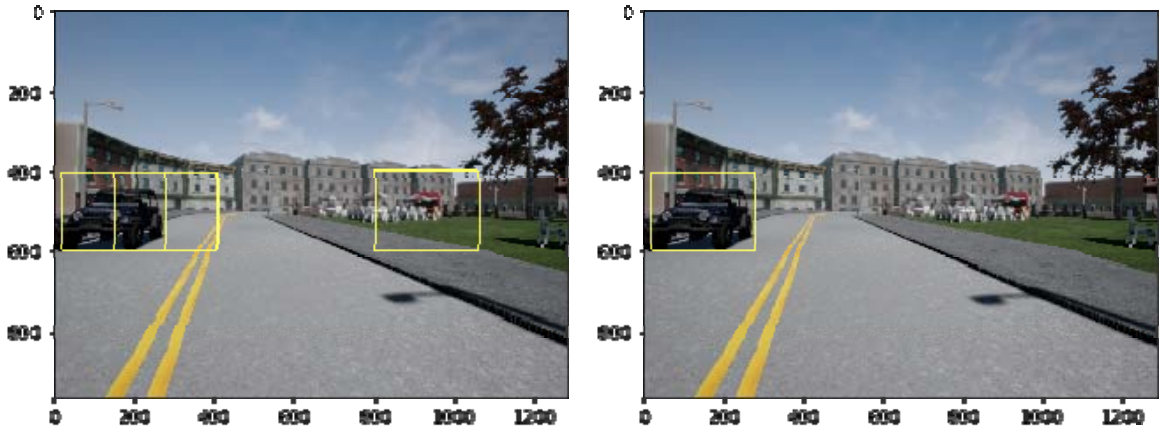

Рисунок 3.19 - Результат візуалізації виявлення вихідних даних зліва та результат фільтрації справа

Результат оцінки відстані до найближчих перешкод на дорозі можливо побачити на рисунку 3.20.

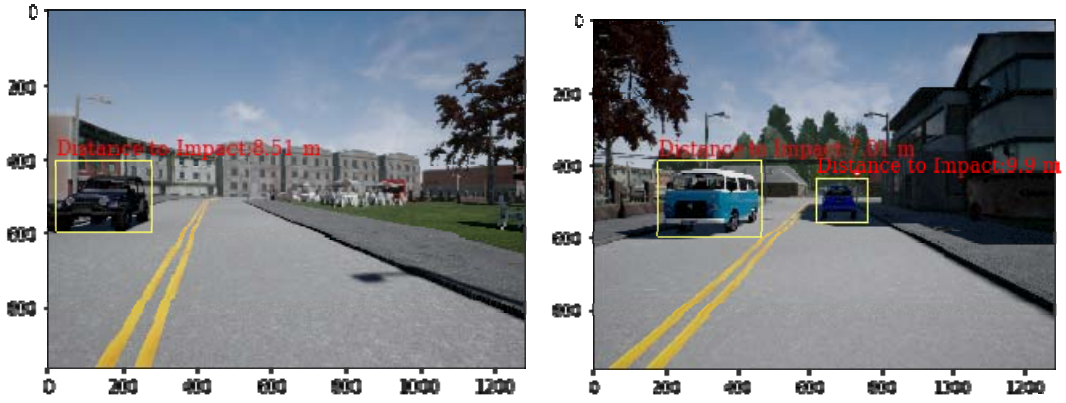

Рисунок 3.20 - Результат оцінки відстані до найближчих перешкод на дорозі

Модель побудови карти середовища була протестована та на базі зроблених тестів була зроблена візуалізація руху автомобіля на виміряній карті переконань. Результати тестування показані на рисунку 3.21.

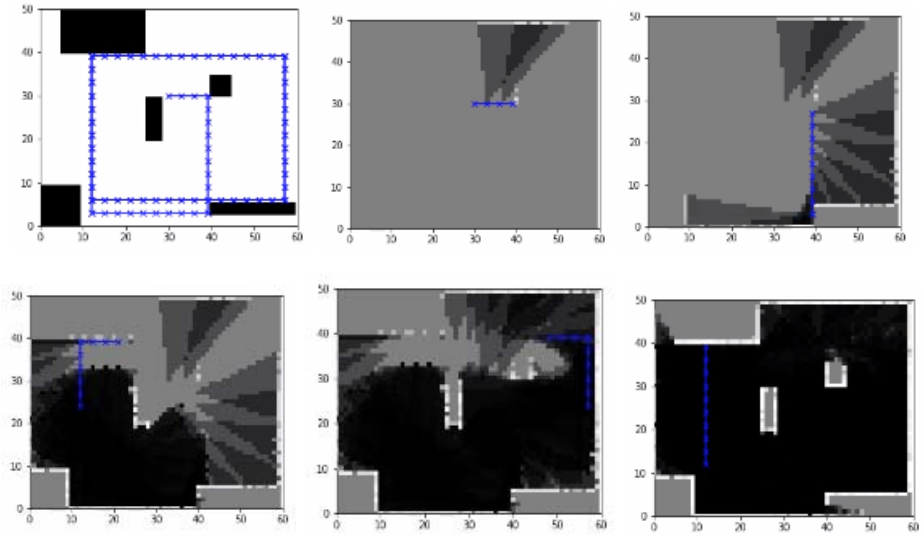

Рисунок 3.21 – Оригінальна карта з перешкодами перший слайд, поетапна візуалізація побудови карти середовища зображення 2-6 слайди

Як можливо побачити в ході тестування модель успішно впоралася з побудовою карти невідомого їй навколишнього середовища. Графічний результат роботи пошуку шляху алгоритмом Дейкстри зліва та алгоритмом пошуку A* справа показаний на рисунку 3.22.

Рисунок 3.22 – Результат пошуку шляху алгоритмом Дейкстри зліва та алгоритмом пошуку A* справа

Опираючись на результат с рисунку вище, оскільки обидва алгоритми показали однаковий результат, для системи був обраний алгоритм Дейкстри, оскільки він більш оптимізований.

Результати роботи нейронних мереж DeepMask, Sharpmask та MultiPathNet можливо побачити на рисунку 3.23.

Рис. 3.23 - Результати роботи нейронних мереж DeepMask, Sharpmask та MultiPathNet

При об’єднанні роботи всіх модулів АСК автомобілем був отриманий результат автоматичної поїдки машини, котрий зображений на рисунках 3.24-3.26.

Рис. 3.24 – Результат автоматичної поїдки машини, начало руху

Рис. 3.25 – Результат автоматичної поїдки машини, поворот

Рис. 3.26 – Результат автоматичної поїдки машини, продовження руху

ВИСНОВКИ

В ході виконання даної дипломної роботи були проаналізовані та досліджені існуючі алгоритми управління АСК автомобілем: ПІД, лінійні та нелінійні регулятори, алгоритм Калмана та його варіації. Серед яких для реалізації АСК автомобілем був вибраний розширений фільтр Калмана за станом помилки.

Вибрані відповідні датасети для навчання нейронних мереж, а саме датасет KITTI Vision для тренування та зображення з CARLA симулятору для тестування розробленої моделі АСК автомобілем.

Були проаналізовані та досліджені існуючі нейронні мережі для рішення питань сегментації, такі як: повністю згорткові мережі, конволюційні моделі з графічними моделями, моделі на основі R-CNN та інші моделі. Серед яких були вибрані наступні нейроні мережі для моделі, котрі показали кращі результати для вирішення даної постановки задачі: DeepMask, Sharpmask та MultiPathNet.

Була розроблена модель АСК автомобілем та визначені математичні моделі підсистем і елементів системи управління автомобілем. Реалізована АСК автомобілем з урахуванням відсутності фізичного обладнання, а саме були реалізовані наступні блоки моделі: картографування, сприйняття, планування та контролеру.

Були успішно проведені дослідження роботи АСК автомобілем в імітованому міському середовищі з будівлями, пішоходами, автомобілями та іншими об'єктами, а також зі змінною погодою, де автомобілем для досліджень була модель звичайного чотиримісного легкового автомобіля, а управління здійснювалось використовуючи правила правостороннього руху.

На базі отриманих результатів був зроблений висновок, що дана АСК автомобілем в подальшому може буди удосконаленою шляхом проведення експериментів в реальному світі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bartels A., Eberle U., Knapp A. Automated Driving Applications and Technologies for Intelligent vehicles, 2015. URL: https://www.adaptive-ip.eu/ (дата звернення: 29.10.2020).

2. Elaine L. Chao. National Highway Traffic Safety Administration URL: https://www.nhtsa.gov/ (дата звернення: 20.10.2020).

3. Learner H. Autonomous Vehicles: Ways to Improve Safety and Accelerate Environmental Progress Together, 2016. URL: http://elpc.org/tag/autonomous-vehicles/ (дата звернення: 23.10.2020).

4. Automated and Autonomous Driving, Regulation under Uncertainty, Corporate Partnership Board Report, 2015. URL: www.internationaltransportforum.org (дата звернення: 25.10.2020).

5. Bailey N. R., Scerbo M. W. Automation-induced complacency for monitoring highly reliable systems: the role of task complexity, system experience, and operator trust, 2007. 321–348.

6. Beggiato M., Krems J. F. The evolution of mental model, trust and acceptance of adaptive cruise control in relation to initial information. Transportation Research Part F: Traffic Psychology and Behaviour, 2013. 47–57.

7. Davis F. D., Bagozzi R. P., Warshaw P. R. User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. Management Science, 1989. 982–1003.

8. Endsley M. R., Kiris E. O. The out-of-the-Loop performance problem and level of control in automation. Human Factors, 1995. 381–394.

9. Ghazizadeh M., Lee J. D., Boyle L. N. Extending the Technology Acceptance Model to assess automation. Cognition, Technology, 2012. 39–49.

10. Hodson H. Driverless cars in gridlock. New Scientist, 2015. 20–21.

11. Hoff K. A., Bashir M. Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust. Human Factors, 2015. 407–434.

12. Kyriakidis M., Happee R. J. Public Opinion on Automated Driving: Results of an International Questionnaire Among 5,000 Respondents, 2014. 127-140.

13. Molloy R., Parasuraman R. Monitoring an automated system for a single failure: Vigilance and task complexity effects. Human Factors, 1996. 311–322.

14. Payre W., Cestac J., Delhomme P. Intention to use a fully automated car: Attitudes and a priori acceptability, 2014. 252–263.

15. European Roadmap Smart Systems for Automated Driving. EPoSS. 2015.

16. Pettersson I., Karlsson I. C. M. Setting the stage for autonomous cars: a pilot study of future autonomous driving experiences, 2015. 694–701.

17. Venkatesh V., Morris M. User Acceptance of Information Technology: Toward a Unified View. MIS Quarterly, 2003. 425–478.

18. Minaee S., Boykov Y., Porikli F., Plaza A., Kehtarnavaz N., Terzopoulos D. Image Segmentation Using Deep Learning. 2020.

19. Markoff J. Google’s Next Phase in Driverless Cars: No Steering Wheel or Brake Pedals. The New York Times. 2014.

20. Lassa, Todd. The Beginning of the End of Driving. Motor Trend. 2014.