

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: «Розробка системи керування безпілотним автомобілем на основі нейронної мережі»

Виконав(ла): студент(ка) VI курсу, групи СНм-61
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Ліщук О.О.

(прізвище та ініціали)

Керівник

(підпис)

Мацюк О.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Мацюк О.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Михалик Д.М.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних наук

Освітній рівень Магістр

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

«____» _____ 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня _____ Магістр
(назва освітнього ступеня)

за спеціальністю _____ 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту _____ Ліщуку Олександр Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи керування безпілотним автомобілем на основі нейронної мережі

Керівник роботи _____ Мацюк О.В. к.т.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом по університету від «28» жовтня 2021 року № 4/7-908

2. Термін подання студентом роботи 20.12.2021 року

3. Вихідні дані до роботи Науково-технічні публікації про моделі нейронної мережі для навчання та керування безпілотним автомобілем.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Вступ. 1 Аналіз науково-технічних публікацій. 2 Моделювання глибокої нейронної мережі.
3 Охорона праці та безпека в надзвичайних ситуаціях. Висновки. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1 Титульна сторінка. 2 Тема, мета, об'єкт, предмет дослідження. 3 Завдання дослідження.
4 Актуальність дослідження. 5 Глибоке навчання безпілотного автомобіля. 6 Навчання нейронної мережі. 7 Транспортний засіб в середовищі Unity. 8 Результати навчання моделей.
9 Результати навчання моделей. 10. Висновки 11. Завершальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання идав	завдання прийняв
Охорона праці	Дмитроца Л.П.		
Безпека в надзвичайних ситуаціях	Клепчик В.М.		

7. Дата видачі завдання: 27 вересня 2021 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	27.09.2020-29.09.2020	<i>Виконано</i>
2.	Підбір наукових джерел про застосування нейронних мереж для навчання та керування безпілотним авто, існуючі нейронні мережі для автономного авто.	30.09.2020-03.10.2020	<i>Виконано</i>
3.	Переклад та опрацювання наукових джерел про застосування нейронних мереж для навчання та керування безпілотним авто	04.10.2020-10.10.2020	<i>Виконано</i>
4.	Виконання дослідження щодо існуючих нейронних мереж для навчання та керування автономного автомобіля	11.10.2020-17.10.2021	<i>Виконано</i>
5.	Оформлення розділу «Аналіз науково-технічних публікацій»	18.10.2021-24.10.2021	<i>Виконано</i>
6.	Оформлення розділу «Моделювання глибокої нейронної мережі»	25.10.2021-31.10.2021	<i>Виконано</i>
7.	Виконання завдання до підрозділу «Охорона праці»	08.11.2021-11.11.2021	<i>Виконано</i>
8.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	12.11.201-14.11.2021	<i>Виконано</i>
9.	Оформлення кваліфікаційної роботи	15.11.201-24.11.2021	<i>Виконано</i>
10.	Нормоконтроль	25.11.2021-28.11.2021	<i>Виконано</i>
11.	Перевірка на плагіат	29.11.2021	<i>Виконано</i>
12.	Попередній захист кваліфікаційної роботи	07.12.2021	<i>Виконано</i>
13.	Захист кваліфікаційної роботи	20.12.2021	<i>Виконано</i>

Студент

(підпис)

Ліщук О.О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Мацюк О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка системи керування безпілотним автомобілем на основі нейронної мережі // Кваліфікаційна робота освітнього рівня «Магістр» // Ліщук Олександр Олегович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2021 // Ст. 88. , рис. – 25 , табл. – 4 , кресл. – 11 , додат. – 1 , бібліогр. – 73.

Ключові слова: нейрона мережа, глибоке навчання, безпілотне авто, J-Net, AlexNet, PilotNet.

В першому розділі кваліфікаційної роботи проаналізовано науково-технічні публікації про методи навчання безпілотного автомобіля за допомогою нейронної мережі.

В другому розділі кваліфікаційної роботи розглянуто вже існуючі моделі нейронних мереж автономних автомобілів. Запропоновано J-Net, модель нейронної мережі яка використовує менше ресурсів для навчання та керування авто. Проведено аналіз результатів J-Net та інших, вже існуючих, моделей нейронної мережі.

В третьому розділі кваліфікаційної роботи розглянуто охорону праці на виробництві автомобільного транспорту та безпеку в надзвичайних ситуаціях.

Основні результати: запропоновано модель нейронної мережі ,J-Net, яка є менш ресурсозатратною і, в порівнянні з іншими дослідженими в цій роботі моделями, показує схожі, а в деяких випадках кращі, навчання та керування авто результати в віртуальному середовищі Unity.

ANNOTATION

Control system development for unmanned vehicle on the base of neural networks// Thesis of educational level "Master" // Lishchuk Oleksandr Olegovich // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, group SNm-61 // Ternopil, 2021 // Article 88., pic. - 25, table - 4, draw. - 11, add. - 1, bibliog. - 73.

Keywords: neural network, deep learning, unmanned vehicle, J-Net, AlexNet, PilotNet.

The first section of the thesis analyzes scientific and technical publications on methods of teaching a drone using a neural network.

In the second section of the thesis existing models of neural networks of autonomous cars are considered. J-Net, a neural network model that uses fewer resources for learning and driving is proposed. The results of J-Net and other existing neural network models were analyzed.

The third section of the thesis deals with labor protection in the production of motor transport and safety in emergencies.

Main results: a neural network model, J-Net, is proposed, which is less resource-intensive and, compared to other models studied in this paper, shows similar, and in some cases better, learning and driving control results in the Unity virtual environment.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ НАУКОВО-ТЕХНІЧНИХ ПУБЛІКАЦІЙ.....	11
1.1 Глибоке навчання безпілотного автомобіля	11
1.1.1 Глибоке навчання у сприйнятті	12
1.1.2 Злиття сенсорів.....	14
1.1.3 Глибоке навчання в локалізації	14
1.1.4 Планування на високому рівні.....	16
1.2 Набір даних для самокерованого автомобіля	17
1.2.1 Семантична сегментація.....	18
1.2.2 Алгоритм YOLO	19
1.3 Оптимізація моделі глибокого навчання для кращої продуктивності...	21
1.3.1 Продуктивність.....	24
1.3.2 Модельна обрізка	25
1.3.3 Зменшення розміру моделі.....	26
1.4 Висновки до розділу 1	27
2 МОДЕЛЮВАННЯ ГЛИБОКОЇ НЕЙРОННОЇ МЕРЕЖІ	28
2.1 Нейронна мережа.....	28
2.2 Глибока нейронна мережа для автономного водіння	30
2.2.1 Автономна система водіння	35
2.2.2 Симулятор середовища Unity.....	37
2.2.3 Попередня обробка даних	40
2.2.4 Архітектура J-Net	44
2.3 Реалізації AlexNet та PilotNet, порівняння з J-Net.....	47
2.3.1 Стратегія навчання та настройка	49
2.3.2 Обчислювальна складність.....	53
2.3.3 Результати перевірки в тренажері	56
2.4 Висновки до розділу 2	62

3 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	64
3.1 Організація охорони праці на виробництві автомобільного транспорту	64
3.2 Підвищення стійкості підприємства автомобілебудівної галузі у воєнний час.....	67
ВИСНОВКИ.....	71
ДОДАТКИ	

ВСТУП

Сьогодні часто постає питання безпеки на дорозі, з кожним роком на дорогах стає все менше місця. Згідно статистиці майже кожна сім'я сьогодні має своє власне авто. В таких реаліях глибоко постає питання безпеки на дорогах. Розробка системи для створення самокерованого автомобіля, використовуючи глибоке навчання, може вплинути на цей показник, зробити водіння автомобіля автономним і менш небезпечним.

Щороку на дорожньо-транспортні пригоди припадає 2,2% смертей у світі. Це складає приблизно 1,3 мільйона на рік - 3 287 на день. Крім того, близько 20-50 мільйонів людей щорічно отримують серйозні травми внаслідок автомобільних аварій. Корінь цих нещасних випадків? Людська помилка.

Від відволікаючого водіння, до водіння у нетверезому стані, до безрозсудного водіння, до необережного водіння - одне погане або неухвалене рішення може бути вирішальним між типовим керуванням автомобілем та ситуацією, що загрожує життю. Але, що якби можна було нейтралізувати людську помилку з рівняння.

«Автономні автомобілі більше не належать голлівудським науково-фантастичним фільмам» - Ілон Маск, засновник Tesla Inc. та SpaceX, вважає, що протягом десяти років самокеровані автомобілі будуть такими ж поширеними, як і ліфти.

Галузеві експерти кажуть, що ця технологія зірве та революціонізує майбутнє транспорту. Прототипом для такого авто є модель глибокого навчання, яка дозволить автомобілю самостійно їздити по дорозі в симуляторі водіння, пізніше а реальних умовах. Це проблема контрольованої регресії між кутами повороту автомобіля та зображеннями доріг у реальному часі з камер автомобіля, якщо в авто стимуляторі всі ці далі легко передбачити, то в реальному світі все набагато важче.

Хорошим варіантом є вже готові проекти, такі як AlexNet та PilotNet, які можна використати як готову основу для нашого безпілота.

Об'єкт дослідження – технології керування та навчання безпілотного автомобіля на основі нейронної мережі

Предмет дослідження – розробка нейронної системи навчання та керування автономним авто.

Мета і завдання роботи. *Метою* роботи є розробка системи керування безпілотним автомобілем за допомогою нейронної мережі та її навчання. Для досягнення поставленої мети необхідно вирішити задачі:

- проаналізувати систему автономного водіння і її складові;
- дослідити існуючі моделі нейронних мереж для навчання безпілотним авто;
- дослідити переваги та недоліки які мають існуючі моделі навчання автономного водіння;
- запропонувати модель нейронної мережі «J-Net» на основі існуючих;
- провести тестування в програмному середовищі Unity, де віртуальне авто буде навчатися за допомогою штучних нейронних мереж, скласти результати дослідження та порівняти J-Net з іншими існуючими нейронними системами автономного водіння;

Наукова новизна. Запропоновано використання модель J-Net для навчання автономного авто, що в порівнянні з іншим існуючими моделями системи навчання безпілотних автомобілів витрачає менше часу на навчання та показує хороші результати на тестовому треку, використовуючи менше ресурсів.

Практичне значення одержаних результатів полягає в можливості застосування даної нейронної мережі для навчання та керування безпілотного автомобіля. Також основу моделі J-Net можна використовувати в інших цілях, наприклад: робот-вантажник, робот-доставки, безпілотне таксі тощо.

Апробація результатів. Основні положення роботи були обговорені на ІХ науково-технічній конференції "Інформаційні моделі, системи та технології".

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох тезах ІХ науково-технічній конференції (Див. додатки А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку літератури з 71 найменувань та 1 додатку. Загальний обсяг кваліфікаційної роботи складає 79 сторінки, з них 56 сторінки основного тексту, який містить 25 рисунків та 4 таблиці.

1 АНАЛІЗ НАУКОВО-ТЕХНІЧНИХ ПУБЛІКАЦІЙ

1.1 Глибоке навчання безпілотного автомобіля

Найперший автономний автомобіль, який використовував нейронні мережі для виявлення ліній смуг руху та сегментував землю для розпізнавання та їзди був АЛВІНН, його було створено в 1989 році. Автономний сухопутний транспортний засіб з нейронною мережею. Також відомий як «Великий Скотт». Як бачимо нейронні мережі використовувалися ще в 1989 році. Підхід був "від кінця до кінця": ви подаєте зображення до нейронної мережі, яка генерує кут повороту керма, відносно отриманих даних, дуже примітивно, але це працювало. Сьогодні на дворі 2021 рік і набагато більше можливостей, більше пізнань в області нейронних мереж і автономних авто, які можна буде застосувати в проекті.

Глибоке навчання захопило основні підрозділи автономного водіння. Для цього я покажу всі чотири основи автономного водіння та поясню, як там використовується глибоке навчання:

- **Сприйняття,** Ви знаходите навколишнє середовище та перешкоди.
- **Локалізація,** Ви визначаєте своє положення у світі з точністю 1–3 см.
- **Планування,** Ви визначаєте траєкторію від А до В, використовуючи сприйняття та локалізацію.
- **Контроль,** Ви слідуєте траєкторії, генеруючи кут повороту та значення прискорення.

Як глибоке навчання реалізовано на всіх чотирьох модулях, і якщо ви прагнете працювати з автомобілем, що керує собою, яких навичок вам потрібно навчитися, чи потрібно бути інженером із глибокого навчання для керування таким авто.

1.1.2 Глибоке навчання у сприйнятті

Сприйняття - це перша опора автономного водіння, і, як ви вже здогадалися, існує багато типів глибокого навчання. Кожен студент, який пройде свій перший курс глибокого навчання, почує: «Глибоке навчання використовується в автомобілях, що керують собою, для пошуку перешкод або смуг руху».

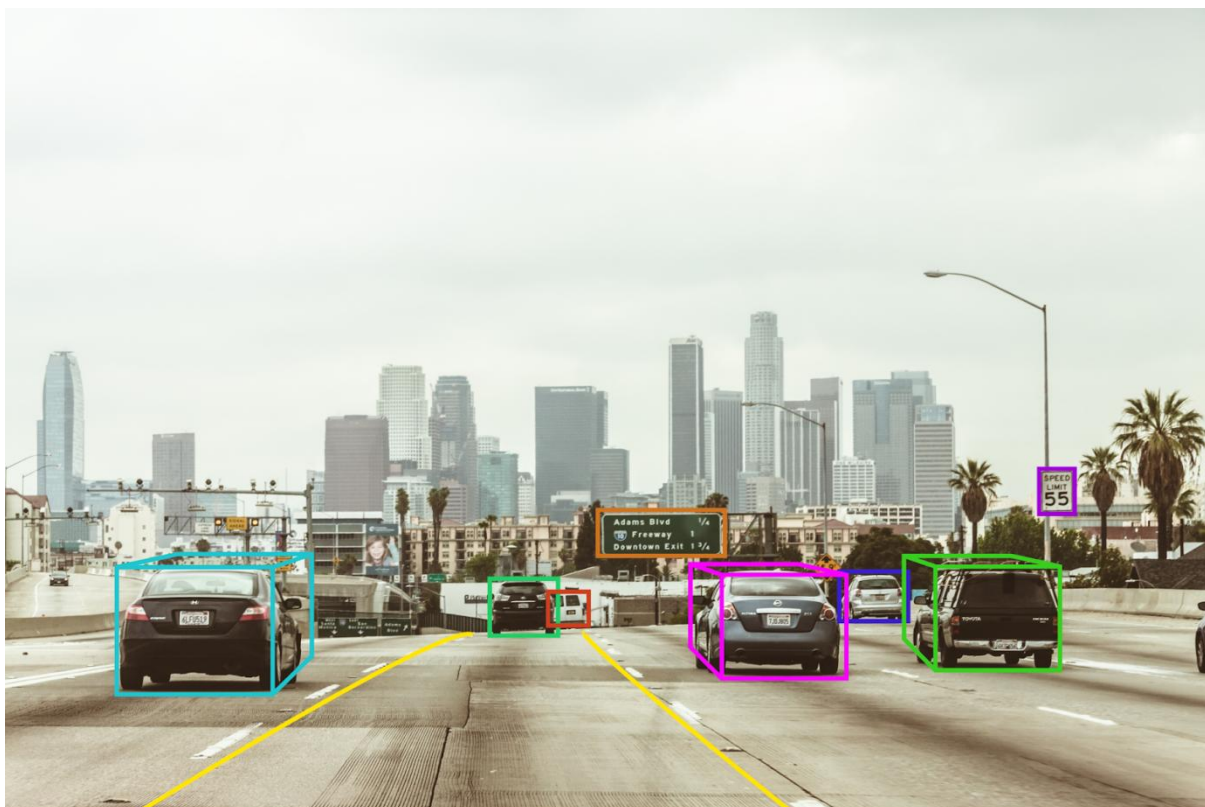


Рисунок 1.1 - Приклад машинного сприйняття

Ці програми належать компанії *Perception*.

Для сприйняття зазвичай використовує 3 сенсори:

- Файл «Камера»
- Файл «ЛіДАР»(Виявлення світла та його діапазону)
- Файл «РАДАР»(Радіовизначення та його діапазону)

На наведеному нижче графіку я показую найважливіші варіанти використання, як вони реалізовані та як реалізовувати це самостійно.

- У світло-жовтому кольорі ви побачите, як вирішувати завдання традиційними методами.

- У темно-жовтому кольорі ви побачите, як вирішувати завдання за допомогою глибоке навчання .
- Синій кольором це рекомендації.

Як бачите, комп'ютерний зір потребує чимало глибокого навчання тільки для першої задачі - виявлення.

Багато налаштувань камери зроблено у стерео. Наявність стерео інформації допомагає нам створити так званий псевдо-LiDAR. Можемо повністю імітувати і навіть іноді замінити LiDAR, а отже, зробити 3D сприйняття за допомогою камер (2D сенсори).

В їх основі знаходимо такі методи, як 3D-CNN (згорткові нейронні мережі) або PointNet - це основа 3D глибокого навчання.

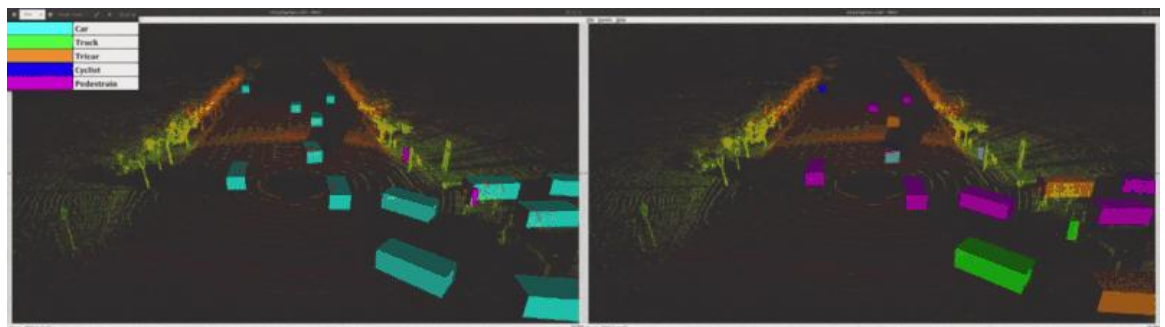


Рисунок 1.2 - Бачення LiDAR

У наші дні виявлення LiDAR за допомогою глибоких нейронних мереж процвітає. Це одне з найактивніших напрямків досліджень у галузі керування автомобілями.

RADAR - дуже давній сенсор. Йому більше 100 років, і не соромно сказати, що для його ефективності не потрібне глибоке навчання. За допомогою RADAR десятиліттями могли вимірювати швидкість об'єкта. Насправді, якщо останнім часом Ви отримали квитанцію за перевищення швидкості, це через RADAR.

Поглиблене навчання в RADAR починає з'являтися за допомогою таких алгоритмів, як центричне 3D виявлення перешкод, або network of region offer RADAR . Однак, схоже, це ще ранні дослідження.

1.1.2 Злиття сенсорів

Заключна частина *сприйняття* - це злиття сенсорів. Щоб зробити виявлення «впевненим», включаємо те, що називається надмірністю. Ідея проста: об'єднуємо дані сенсори і перевіряємо, чи вони показують одне і те саме.

Існує 3 способи об'єднання за допомогою трьох сенсорів:

- Об'єднання камери та LiDAR
- Об'єднання камери та RADAR
- Об'єднання LiDAR та RADAR
- Розрізняють раннє та пізнє злиття(синтез).
- Раннє злиття означає, що ви об'єднуєте необроблені дані, такі як хмари точок LiDAR та пікселі зображення.
- Пізнє злиття означає, що ви поєднуєте вихідні дані виявлення, наприклад, 2D та 3D Bounding Box.

Цікаво, що глибоке навчання більш-менш підходить для синтезованих сенсорів, залежно від використаних сенсорів. Злиття камери і LiDAR один з процесів традиційного підходу.

Глибоке навчання дуже добре підходить для цього процесу. З допомогою глибокого навчання, окремо камеру і LiDAR, використаємо глибоке навчання для їх злиття.

1.1.3 Глибоке навчання в локалізації

Локалізація - це пошук позиції транспортного засобу Ego в світі. Перше, що спадає на думку - це використання GPS, але ви виявите, що він може бути дуже неточним і може не працювати ідеально кожного разу, наприклад, коли хмарно.

Зрештою, GPS точний до 1-2 м. Ця проблема створила ціле поле, яке назвемо локалізацією. Залежно від вибору алгоритму, є багато способів локалізації:



Рисунок 1.3 - Вигляд з висоти пташиного польоту (Birdeye)

Тепер уявіть, що ви все ще перебуваєте в Нью-Йорку, але вас викрали, зав'язали очі та помістили кудись подальше. Щоб визначити свою позицію, вам потрібно буде використати свої очі та знання карти. Коли ви впізнаєте щось знайоме, наприклад Емпайр-Стейт-Білдінг, ви дізнаєтесь, де ви знаходитесь.

Для цього використовуємо розширені фільтри Калмана та фільтри частинок. Як ви помітили, також використовуємо Odometry (кількість обертів колеса, GPS, GPS-RTK (кращий GPS) і UWB (трилатерація за допомогою фізичних пристроїв). Якщо ви шукаєте глибоке навчання, то тут є лише знакове виявлення, яке отримується на етапі сприйняття.

Поле SLAM спочатку використовує багато Байєсівської фільтрації, наприклад, фільтри Калмана та частинок, але наразі процвітає те, що називається візуальною одометрією.

Як бачимо, для картографування та локалізації залучено багато поглибленого навчання, але ця сфера переважно не використовує глибоке навчання. Якщо ви хочете бути інженером з локалізації, набагато важливіше добре розуміти робототехніку та традиційні методи, ніж глибоке навчання.

1.1.4 Планування на високому рівні

Планування - це мозок автономного транспортного засобу. Він проходить шлях від передбачення перешкод до створення траєкторії. По суті: прийняття рішень.

Можемо поділити світ планування на 3 кроки:

- Глобальне планування високого рівня - програмування маршруту від А до В.
- Поведінкове планування - передбачення того, що будуть робити інші транспортні засоби, перешкоди, та інші об'єкти на дорозі.
- Шлях/місцеве планування - уникайте перешкод і створіть траєкторію.

Зазвичай використовується A^* . Але ви також знайдете багато глибокого навчання з посилення: це називається ймовірнісним плануванням.

Цей крок включає 2 підкроки:

- Передбачення
- Прийняття рішень

У прогнозуванні потрібно використовувати інформацію про час та асоціацію даних, щоб зрозуміти, де перешкода буде в майбутньому. Можна знайти підходи, що ґрунтуються на навчанні, такі як моделі суміші Гауса для передбачення намірів, і також підхід фільтра Калмана; Комп'ютерне бачення для відстеження.

Прийняття рішень - це щось інше. Або вводимо деякі правила вручну, і створюємо кінцевий стан машини, або не робимо цього, і використовуємо підходи з підкріпленням навчання.

Глобальне планування - це знання того, де ви повинні їхати на карті, маршрут. Але що робити, якщо автомобіль блокується. Що робити, якщо світлофор світиться червоним. Що якщо автомобіль дуже повільний. Нам потрібно щось зробити, наприклад змінити траєкторію руху або зупинити автомобіль.

Висновок, як бачите, глибоке навчання добре застосовується у багатьох сферах автономного водіння, і воно поступово з'являється у всіх інших.

1.2 Набір даних для самокерованого автомобіля

Побудова проектів самокерованих автомобілів – насправді, в сьогоdnішніх реаліях, просто. За ці роки багато було зроблено для надання відповідної допомоги інженерам з глибокого навчання, які хочуть навчати моделі безпілотних автомобілів. Проблема даних довгий час була величезною. Спеціальні набори даних, он-лайн інструменти для створення анотацій - все було розроблено, щоб допомогти з Computer Vision.

Почнемо з міток зображень, а потім розглянемо решту. Перші три папки - "Зображення", "Сегментація" та "Карти для керування" - містять зображення та зображення сегментації.

Сегментація - це тип алгоритму, метою якого є призначення класу кожному пікселю шляхом створення карти. Таким чином, мітка - це зображення, де кожен піксель складається з одного кольору, що представляє клас.

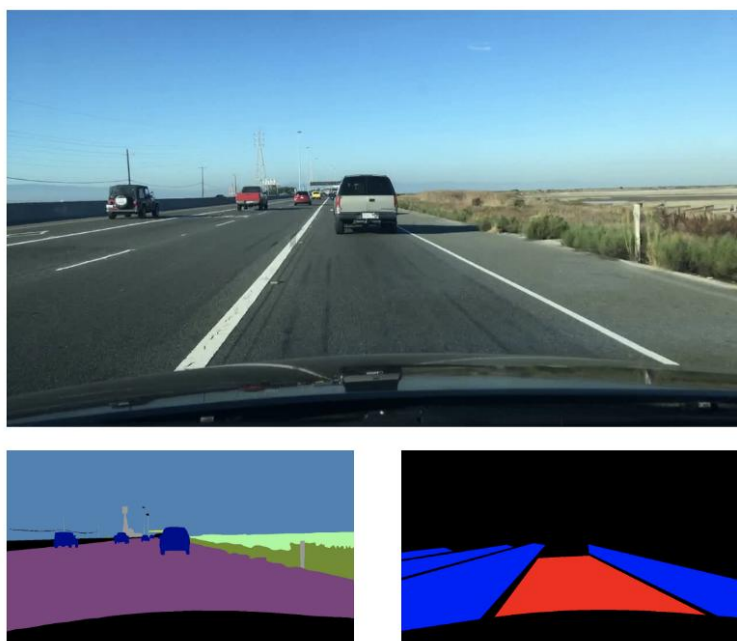


Рисунок 1.4 – Зображення з мітками, сегментація областей з BDD100K

- Зображення є зображенням 1280x720 RGB. Назва файлу зображення - це певний номер, який індивідуальний для кожного зображення.
- У семантичній сегментації кожен піксель є одним об'єктом. Тут тип сегментації є повнокадровою сегментацією. Таким чином, ви можете бачити дорогу фіолетовим кольором, небо голубим, автомобілі синім (див. Рисунок 1.9)
- І останнє, карти для керування - це смуги руху, червоні пікселі - смуга Ego, де можна їздити, сині пікселі позначають сусідні смуги (див. Рисунок 1.9). Для семантичної сегментації є багато класів, основні – це «проїжджа» і «не проїжджа» дорога.

1.2.1 Алгоритми семантичної сегментації

Щоб використовувати дані такого типу, вам знадобляться алгоритми семантичної сегментації. Зазвичай це робиться за допомогою кодера/декодера. Карти сегментації дають нам багато інформації, але, ймовірно, недостатньо. Зокрема, було б непогано мати обмежувальні рамки кожного об'єкта. Завдяки цьому зможемо запустити такі алгоритми виявлення перешкод, як YOLO, FASTER RCNN.



Рисунок 1.5 – Семантична сегментації

Перша частина містить інформацію, що стосується лише місця події загалом. Ось розподіл набору даних для цих елементів. Далі отримаємо доступ до маркування перешкод.

Друга частина дуже допоможе у виявленні перешкод. Кожна перешкода повністю описується обмежувальною рамкою. Кожна перешкода має однакову інформацію. Інструмент маркування використовувався для того, щоб вручну намалювати обмежувальну рамку та додати інформацію до файлу.



Рисунок 1.6 – Зображення робочого простору BDD 100K

Як бачите (Рисунок 1.6), інформацію про JSON можна подати за допомогою інструменту маркування. Ось розподіл перешкод для цього набору даних. Для використання такого типу інформації вам знадобиться алгоритм виявлення перешкод, такий як YOLO, SSD або FASTER RCNN.

1.2.2 Алгоритм YOLO

Метою алгоритму буде передбачити координати обмежувальної рамки. Для кожної смуги у файлі існує один абзац. Тут знову натискаємо на точки смуги, щоб мати точне зображення дороги. Кожна точка, яку ви бачите у файлі JSON, є точкою на краю смуги.

Подивимось розмітку смуги руху. Це схоже на маркування зони водіння, але для одного ряду. Вершини представляють рівняння лінії смуги руху.

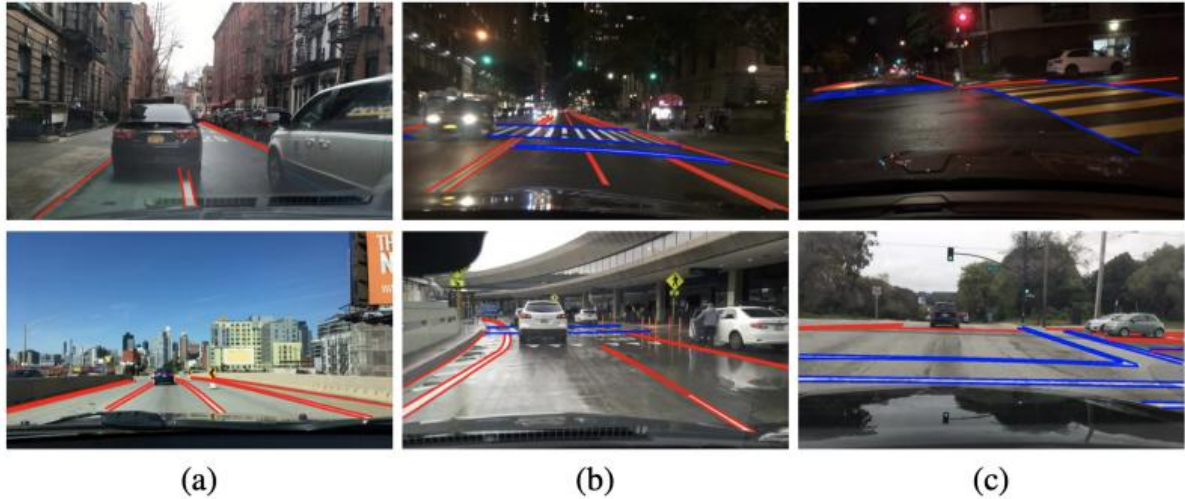


Рисунок 1.7 – Маркування розмітки

Розповсюдження. Ці останні ознаки є актуальними і їх можна використовувати. Наприклад, якщо розмітка смуги паралельна автомобілю, що проходить повз, це може служити прикладом для керування автомобілем та розділення смуг руху; якщо він вертикальний, це можна розглядати як ознаку уповільнення або зупинки.

Щоб використовувати 2D значення, вам знадобиться нейронна мережа, яка може передбачати рівняння лінії смуги. Результатом буде набір чисел, які ви намагаєтесь регресувати. Цифри - це ті, що є у файлі JSON.

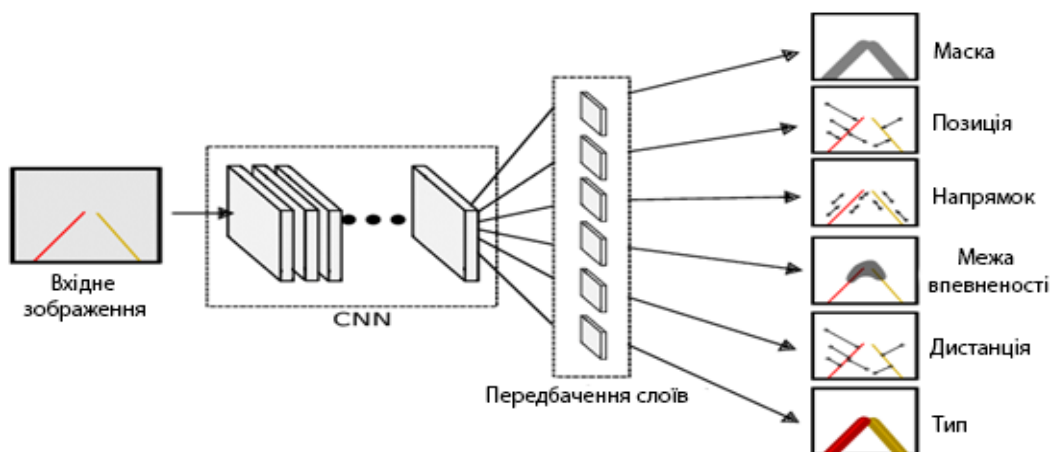


Рисунок 1.8 – LineNet

Лінії смуг також можна передбачити за допомогою семантичної сегментації. Набір даних BDD 100K надзвичайно багатий. Одне, що я не побачив при вивченні файлів JSON, - це 3D обмежувальні коробки, які сьогодні стають все більш актуальними. Навчання підмножині цих зображень дасть вам дуже сильне уявлення про світ. Це зменшує потребу в таких техніках збільшення даних, як перегортання, обрізання, обертання зображення. Те ж саме стосується відео, де цього разу можна отримати доступ до відстеження послідовних кадрів. Інші набори даних, такі як CityScape або Kitty, заповнені наполовину.

1.3 Оптимізація моделі глибокого навчання для кращої продуктивності

Чи є спосіб оптимізувати модель глибокого навчання. Які процеси. Чи легко це зробити. Або це секрет експертів, яких схований від лишніх очей. У глибокому навчанні, висновок - це процес розповсюдження вперед, який, з огляду на вхідні дані, отримує результат. Наприклад, класифікація 3D хмари точок може визначити клас, а регресія може отримати координати обмежувальної рамки перешкод на зображенні.

Наприклад, можемо замінити стандартну згортку на роздільну згортку, зменшивши кількість обчислень. Також можемо обрізати або заморозити модель. Ці методи оптимізації дозволяють заощадити секунди часу висновку та різко змінити результат.

Для того, щоб зрозуміти, як оптимізувати нейронну мережу, потрібно мати метрику. Цей показник буде часом висновку.

Час висновку - це скільки часу потрібно для прямого розповсюдження. Щоб отримати кількість кадрів в секунду, ділимо 1/час висновку.

Щоб виміряти цей час, треба зрозуміти три ідеї: FLOP, FLOPS та MAC.

FLOP: щоб виміряти час висновку для моделі, можемо обчислити загальну кількість обчислень, які модель повинна виконати. Тут згадуємо

термін FLOP або операція з плаваючою комою. Це може бути додавання, віднімання, ділення, множення або будь-яка інша операція, що включає значення з плаваючою комою. FLOP дасть нам понизити складність нашої моделі.

Bookmarks: інакше - це FLOPS із великою буквою S.

Bookmarks - це операції з плаваючою комою за секунду. Це швидкість, яка говорить нам про те, наскільки швидке наше обладнання. Чим більше операцій за секунду зможемо зробити, тим швидше буде висновок.

MAC: останній момент, на який слід звернути увагу - це MAC, що означають множинно-накопичувальні обчислення. MAC - це операція додавання та множення, тобто дві операції.

$$\text{Вихідний рівень} = W1 * A1 + W2 * A2 + \dots + Wn * An \quad (1.1)$$

Як правило, вважаємо $1 \text{ MAC} = 2 \text{ FLOP}$.

Тепер, коли було вивчено загальну ідею:

1. Краще, щоб у нашій моделі була невелика кількість FLOP, але при цьому вона залишалася б достатньо складною, щоб бути точною.
2. Потрібна велика кількість FLOPS у обладнанні.
3. Наша роль буде полягати в тому, щоб оптимізувати моделі глибокого навчання, щоб мати якнайменшу кількість FLOP.

Обчислення FLOP у моделі:

Давайте візьмемо таку модель, яка виконує класифікацію на наборі даних MNIST.

- Вхідне зображення має розмір $28 \times 28 \times 1$ (чорно-біле)
- Запускаємо 2-і згортки з 5-и ядер розміром (3×3)
- Запускаємо повністю з'єднаний шар із 128-ми нейронів
- Закінчуємо повністю з'єднаним шаром з 10-и нейронів: 1 на цифру.

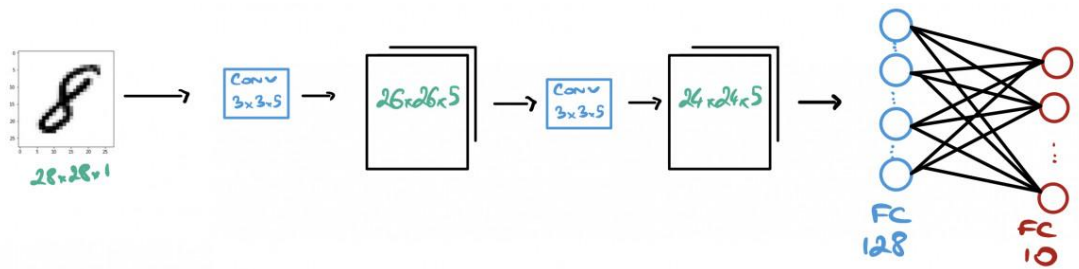


Рисунок 1.9 - Оптимізація навчання за допомогою класифікації MNIST

Кожен рівень цієї моделі буде виконувати деякі операції, які додадуть деякі FLOP.

Для оновлення на CNN можна переглянути цю шпаргалку.

Щоб обчислити FLOP у моделі, ось правила:

Витки FLOP = 2 * кількість ядра * форма ядра * форма виводу

Повністю з'єднані шари FLOP = 2 * розмір входу * розмір виводу

Ці формули фактично створені для розрахунку кількості MAC. FLOP отримуються з «2*» на початку.

Обчислимо це з урахуванням співвідношень:

Перша згортка: $2 * 5 * (3 * 3) * 26 * 26 = 60,840$ FLOPs. Друга згортка: $2 * 5 * (3 * 3 * 5) * 24 * 24 = 259,200$ FLOPS. Перший шар FC: $2 * (24 * 24 * 5) * 128 = 737,280$ FLOPS. Другий шар FC - $2 * 128 * 10 = 2560$ FLOP.

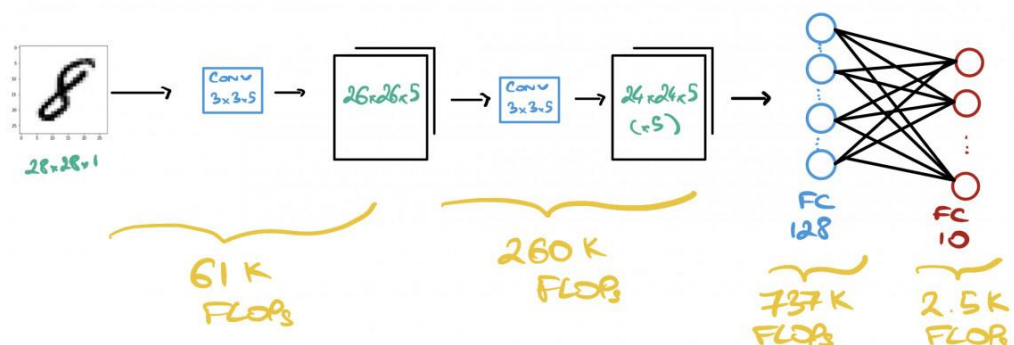


Рисунок 1.10 – FLOP і FLOPS у CNN

Модель буде виконувати операції $FLOP = 60\,840 + 259\,200 + 737\,280 + 2560 = 1060\,400$ операцій.

1.3.1 Продуктивність

Скажімо, є процесор, який виконує 1 GFLOPS. Час висновку буде $FLOPS/FLOPS = (1,060,400)/(1,000,000,000) = 0,001$ с або 1 мс. Обчислення часу висновку просте, якщо є FLOPS. FLOPS можна отримати, розуміючи, що таке наш процесор. Чим потужніший процесор, тим більше це число. Давайте тепер подивимось на другу частину: оптимізація. Як оптимізувати модель для кращої продуктивності.

Тепер подивимось, як оптимізувати модель глибокого навчання для висновків. Це може бути дуже корисним і в інших ситуаціях, наприклад, під час розгортання або просто під час запуску моделі. Маємо два основні способи оптимізації нейронної мережі:

- За рахунок зменшення розміру моделі
- За рахунок зменшення кількості операцій

Коли зменшуємо кількість операцій, замінюємо деякі операції на більш ефективні. Тут дозвольте мені навести три операції, які можуть спрацювати:

- Об'єднання
- Роздільні згортки
- Модельна обрізка

Об'єднання. Почнемо з об'єднання. Якщо ви розумієте що таке глибокого навчання, то, напевно, ви вже знаєте про максимальне або середнє об'єднання. Ці операції є частиною тих, що зменшують кількість операцій.

Об'єднані шари - це шари субвибірки, які зменшують кількість даних або параметрів, що передаються від одного шару до іншого. Шари об'єднання зазвичай використовуються після згортки для збереження просторової інформації, зменшуючи при цьому кількість параметрів.

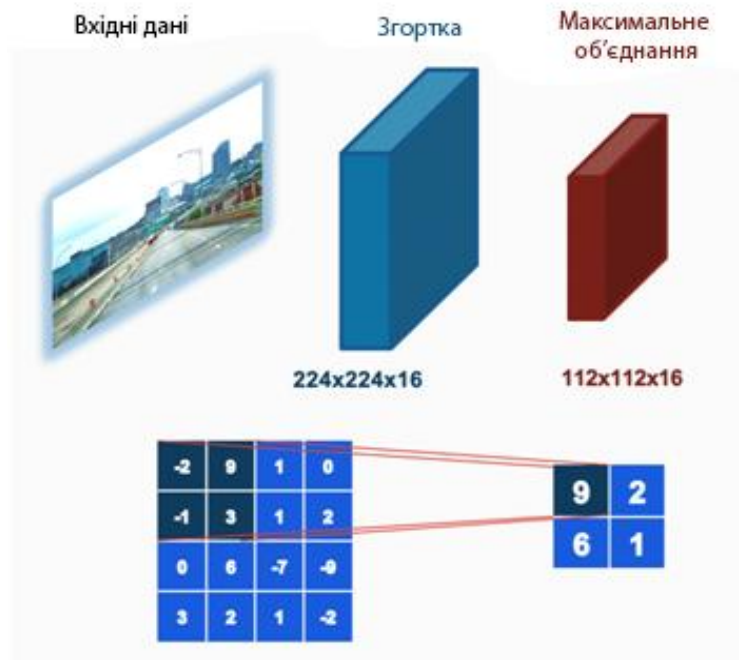


Рисунок 1.11 - Максимальне об'єднання

Об'єднання це чудово, але при розрахунку кількості FLOP це все ще досить затратно.

Роздільна згортка - це згортковий шар, який розділяє стандартний згортковий шар на два шари: глибоку згортку та точкову згортку.

Тепер подивимось на еквівалентну двовимірну згортку:

Понад 20 мільйонів операцій. За допомогою програмного забезпечення FLOPS об'ємом 1 гігабіт ця операція займе 0,2 секунди.

Роздільні згортки можуть різко скоротити кількість операцій. Маємо 20 мільйонів операцій проти 734 тисяч, або більш ніж у 20 разів менше при використанні цих операцій. Це майже різниця 0,2 с за висновок.

1.3.2 Модельна обрізка

Обрізання - це метод стиснення моделі, коли надлишкові параметри мережі видаляються, намагаючись зберегти початкову точність (або інший показник) мережі.

Щоб досягти цієї техніки, спочатку визначаємо найбільш важливі ранги, а потім видалимо зайві. Для цього або видалимо нейрон, встановивши для нього з'єднання 0 або встановивши вагу на 0 .

Щойно проаналізували три потужні методи оптимізації глибокої нейронної мережі за рахунок зменшення кількості операцій. Зверніть увагу, що їх можна легко реалізувати одним рядком коду

1.3.3 Зменшення розміру моделі

Тепер подивимося, як зменшити розмір моделі. Коли зменшуємо розмір моделі, отримуємо ряд переваг, таких як швидше завантаження, менший розмір сховища, більш швидка компіляція.

Є три способи зменшення розміру моделі:

- Квантування
- Перегонка знань
- Розподіл ваги

Квантування

Почнемо з квантування моделі, яке є процесом стиснення.

Квантування - це процес зіставлення значень з більшої множини на меншу. Іншими словами, зменшуємо великі безперервні числа і замінюємо їх меншими безперервними чи навіть цілими числами.

Якщо вага 2.87950387409, стисненою вона буде 2.9. Тому змінюється точність.

Кілька речей, які слід зазначити щодо квантування.

- Квантування можна проводити як на вагах, так і на активностях.
- Це зменшує об'єм, але ускладнює обчислення.

Загалом, будемо використовувати це для нормалізації ваг або зміни діапазону ex: перетворення ваг від -3 до +5 до діапазону 0-255.

Квантування належить до сімейства, яке називається компресією. Розподіл ваги - це ще один метод, за допомогою якого розподіляємо вагу між нейронами, тому їх менше для зберігання.

1.4 Висновки до першого розділу

В цьому розділі вивчено основи автономного водіння, типи комп'ютерного бачення та способи їх оптимізації. Чотири стовпи на яких тримається безпілотне керування це:

- Сприйняття
- Локалізація
- Планування
- Контроль

Ознайомились з видами бачення, та обробки інформації, які використовуються для управління безпілотним автомобілем. Такі як RADAR і LiDAR які є головними для орієнтації автомобіля в просторі.

Розуміння BDD 100K набору даних, це вже готова система в якій є готові, оброблені зображення для навчання нейронної мережі. Також дізнались ,як система виявляє та розпізнає перешкоди, отримали основні знання для розуміння алгоритму YOLO, та його властивостей.

Вкінці обговорили оптимізацію, її плюси та мінуси. Перечисли основні методи оптимізації і показали на прикладі, як правильно оптимізувати модель навчання нейронної мережі.

2 МОДЕЛЮВАННЯ ГЛИБОКОЇ НЕЙРОНОХ МЕРЕЖІ

2.1 Нейронна мережа

У цьому проєкті був використаний симулятор водіння, основою якого є високорівнева системи збору даних Nvidia - Udacity, який має два різні напрямки. Один з них використовувався для збору навчальних даних, а інший – для нових, раніше опізнаних моделей - як заміна тестовому набору. Симулятор водіння зберігатиме кадри з трьох фронтальних «камер», записуючи дані з точки зору автомобіля; а також різні статистичні дані про їзду, такі як дросельна заслінка, швидкість і кут повороту. Використовуємо дані камери як вхідні дані для моделі та очікуємо, що вона передбачить кут повороту в діапазоні $[-1, 1]$.

Навколишнє середовище та інструменти

1. matplotlib
2. keras
3. numpy
4. panda
5. scikit-learn

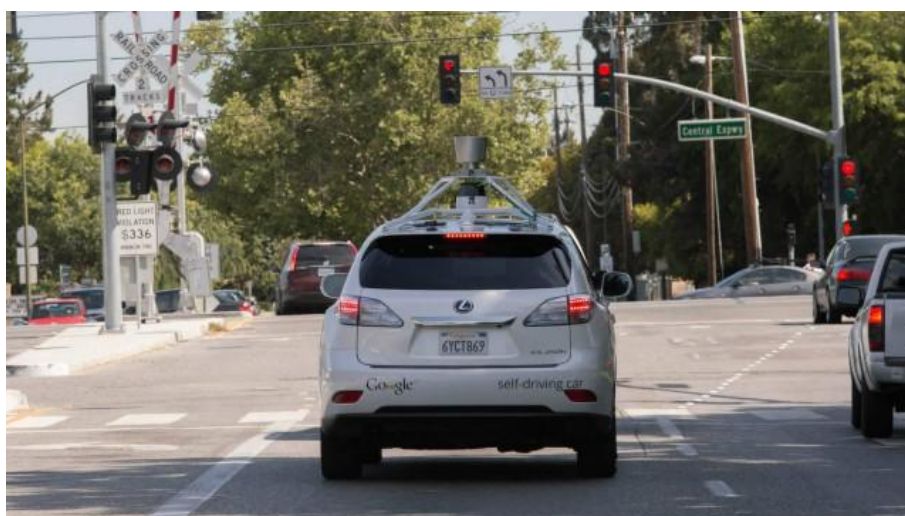


Рисунок 2.1 - Самокерований автомобіль Google-Waymo

Набір даних містить шість рядків – камери по центру, ліворуч, праворуч (шляхи зображення камери), рульове управління, дросельна задня передача - задній хід, швидкість. Я використовував pandas для відображення перших п'яти рядків у наборі даних.

Оскільки префікс лівого, правого та центрального шляхів зображення був однаковим для всіх рядків, я вирішив видалити частину префіксу у всьому наборі даних.

Далі я побудував графік розподілу значень кута нахилу керма. Як бачимо, біля нуля існує величезний шип (див. Рисунок 2.15), що означає, що в більшості випадків автомобіль їде прямо. Потім я зробив функцію завантаження всіх зображень, а також додав значення кута керма в масив numpy.

Наступним кроком було розбиття даних за правилом 80–20, що означає використання 80% даних для навчання, а решта - для тестування моделі на раніше не оброблених (небачених) зображеннях. Крім того, я побудував графік розподілу кутів рульового навчання та валідації.

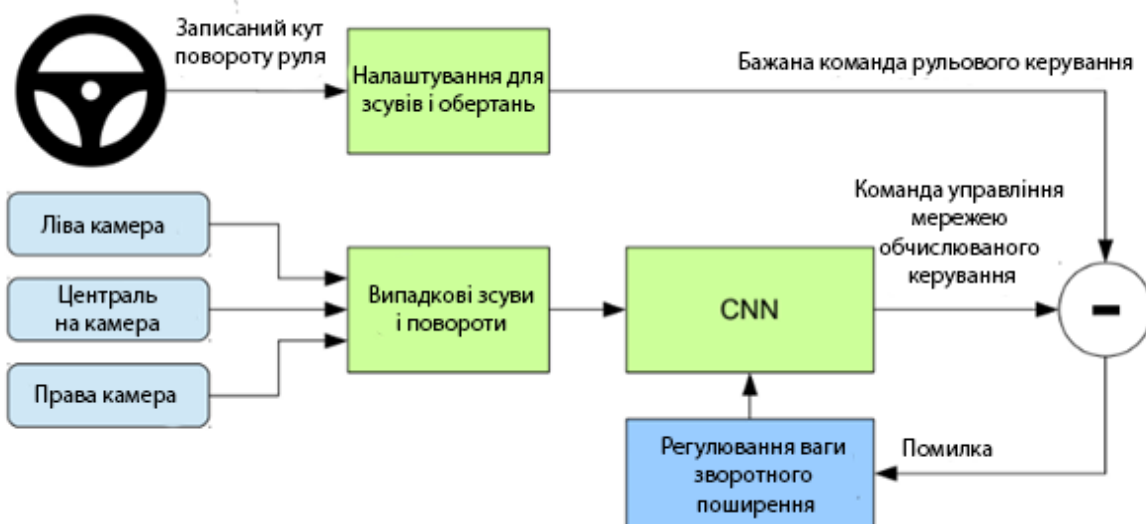


Рисунок 2.2 - Навчання нейронної мережі

На додаток до важкої архітектури resnet, я використав вирівняний шар для нормалізації ваг. Далі я використав три щільні шари із 100, 50 та 10

нейронами відповідно та elu як функцію активації. Крім того, я використовував 50% dropouts, щоб зменшити надто відповідні значення навчальному набору.

Нарешті, я навчив модель 25-ти епох з розміром партії 128. Крім того, я побудував графік навчання та втрати валідації як функцію епох.

Модель досить добре збігається всього за 25 епох. Це означає, що керувати автомобілем у невидимих дорожніх умовах навчається досить добре. Не соромтеся грати з гіперпараметрами для кращих результатів.

Результати. Я був здивований тим, наскільки добре машина їздила на тестовій трасі. CNN ніколи не бачив цього треку. Виступ на тренувальному треку був трохи невдалим, але я думаю, що це нормально, оскільки це показує, що машина не просто запам'ятовувала трек. Він успішно реабілітувався після кількох критичних ситуацій, хоча жоден з цих маневрів не був виконаний під час навчання.

2.2 Глибока нейронна мережа для автономного водіння

Дослідження та розробки у сфері машинного навчання, а точніше - глибокого навчання, призводять до багатьох відкриттів та практичних застосувань у різних сферах. Область, де машинне навчання має величезний вплив - це автомобільна промисловість та розвиток повністю автономних транспортних засобів. Рішення машинного навчання використовуються в кількох автономних підсистемах транспортних засобів, як сприйняття, злиття сенсорів, одночасна локалізація та відображення, а також планування шляху, про це говорилось в першому розділі. Паралельно з роботою по повній автономії комерційних автомобілів поточною тенденцією є розвиток різних автомобільних платформ. Наприклад, на складах використовуються транспортні засоби доставки або різні роботи та автомобілі-роботи. Основною ідеєю нашої роботи було розроблення рішення для автономного водіння для легкої автомобільної платформи, що має обмежені апаратні

ресурси, потужність процесора, і обсяг пам'яті. Маючи на увазі ці апаратні обмеження, прагненням є: спроектувати легку глибоку нейронну мережу (DNN), наскрізну нейронну мережу, яка зможе виконувати завдання автономного водіння на репрезентативній трасі, тоді як розвинені мережі модель, яка використовується для висновків, можна розгорнути на низькопродуктивній апаратній платформі.



Рисунок 2.3 – Структурна схема системи автономного транспортного засобу

Наскрізна глибока нейронна мережа, яку я розробив для автономного водіння, використовує зображення камери як вхідний сигнал, який є необробленим сигналом (тобто пікселі), і прогнозування кута повороту як вихід для управління транспортним засобом. Наскрізне навчання - це навчання нейронних мереж від початку до кінця без людської взаємодії або участі в навчальному процесі. Метою наскрізного навчання є те, що система автоматично вивчає внутрішні уявлення про необхідні етапи обробки, такі як виявлення корисних характеристик дороги, базуючись лише на вхідному сигналі.

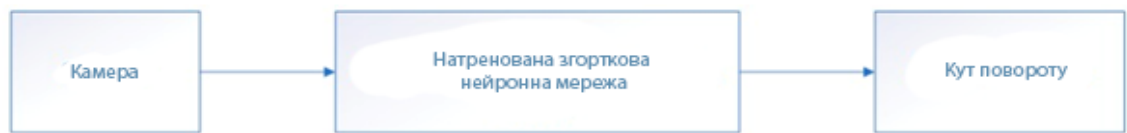


Рисунок 2.4 - Структурна схема наскрізної системи автономного водіння.

В наші дні програми машинного навчання все частіше розгортаються на вбудованих пристроях, мобільних телефонах та рішеннях Інтернету речей (IoT). Розгортання рішень машинного навчання на вбудованих апаратних платформах призводить до нових розробок у двох напрямках: розробка нових апаратних платформ, здатних обробляти дані, необхідні для машинного навчання, та розробка нових архітектур легкого машинного навчання та реалізації моделей, придатних для низькопродуктивного обладнання.

Відомі рішення для наскрізного навчання для автономного водіння [1,2,3,4,5] розроблені здебільшого для реальних транспортних засобів, де модель машинного навчання, що використовується для висновків, розгортається на високопродуктивному комп'ютері, який зазвичай розташований у багажнику транспортного засобу, або ці рішення використовують дуже глибокі нейронні мережі, які обчислювально дорогі (наприклад, використовуючи архітектуру ResNet50) [3]. Однак наша ідея полягала в тому, щоб розробити значно менше рішення, легку глибоку нейронну мережу, з такою ж продуктивністю під час автономного керування, але з використанням менших обчислювальних витрат, які дозволять розгортання на вбудованій платформі. Це більш легке рішення буде використовуватися для автомобілів-роботів, вбудованих автомобільних платформ, здатних перевозити вантажі або виконувати деякі подібні завдання серед відносно відомих траєкторій.

У цій роботі я представляю нову глибоку нейронну мережу, розроблену з метою наскрізного навчання для автономного водіння, під назвою J-Net, яка призначена для вбудованих автомобільних платформ. На додаток до цього, з метою об'єктивної оцінки J-Net, обговоримо результати нашої повторної реалізації PilotNet [1,2] та AlexNet [6]. AlexNet спочатку розроблений для

класифікації об'єктів, але після наших модифікацій нова модель, схожа на AlexNet, підходить для автономного водіння. По-перше, розробляється нова архітектура глибокої нейронної мережі J-Net, модель навчається та використовується для висновків під час автономного водіння. По-друге, було порівняно архітектури трьох мережевих архітектур, J-Net, PitotNet та AlexNet, та обговорили обчислювальну складність. Це робиться з причини об'єктивної оцінки архітектури нашої мережі. Далі впроваджені моделі пройшли навчання з тим самим набором даних, який було зібрано. Збір даних та висновки здійснюються в симуляторі автономного автомобіля, розробленому в середовищі Unity компанією Udacity, Inc. [7]. Нарешті, навчені моделі були використані для автономного водіння в режимі реального часу в середовищі тренажера. Були представлені та порівняні результати автономного водіння з використанням кожної з цих трьох моделей глибоких нейронних мереж. Результати автономного водіння подаються у вигляді відеозаписів автономного водіння на репрезентативній доріжці в середовищі симулятора, на додаток до якісної та кількісної оцінки параметрів автономного водіння під час висновку.

Результати показують, що хоча складність і розмір нової мережі менші в порівнянні з іншими моделями, J-Net підтримує продуктивність, досягаючи подібної ефективності в автономному водінні. Прагнення до впровадження на вбудованій автомобільній платформі підсилює важливість обчислювального легкого рішення для DNN, що використовується для автономного водіння, оскільки вбудовані системи можуть постраждати від апаратних обмежень для бортових комп'ютерів, які не здатні виконувати найсучасніші технології глибокого навчання моделей.

Система автономного водіння, що використовується для збору даних та автономного водіння, набір даних та стратегія збору даних наведені в першому розділі. Запропонований підхід, архітектура та деталі реалізації J-Net наведені у другому розділі. Реалізація моделей, порівняння архітектури та параметрів мережі та стратегія навчання для всіх трьох рішень

представлені у кінці другого розділу. Результати та обговорення впровадження всіх трьох нейронних мереж та висновки під час автономного керування наведені у висновку до другого розділу та у загальному висновку.

Згорткові нейронні мережі (CNN) - це спеціалізований вид нейронної мережі для обробки даних, що має відому сітчасту топологію. CNN об'єднують три архітектурні ідеї: місцеві репрезентативні поля, спільні ваги та просторову чи часову підвибірку, що призводить до певної міри зсуву, масштабу та інваріантності спотворень. Згорткові нейронні мережі призначені для обробки даних з декількома масивами (наприклад, кольоровим зображенням, мовою, аудіоспектрограмою та відео) та отримують вигоду від властивостей таких сигналів: локальних з'єднань, загальних ваг, об'єднання та використання багатьох шарів. З цієї причини CNN найчастіше застосовуються для аналізу візуальних зображень.

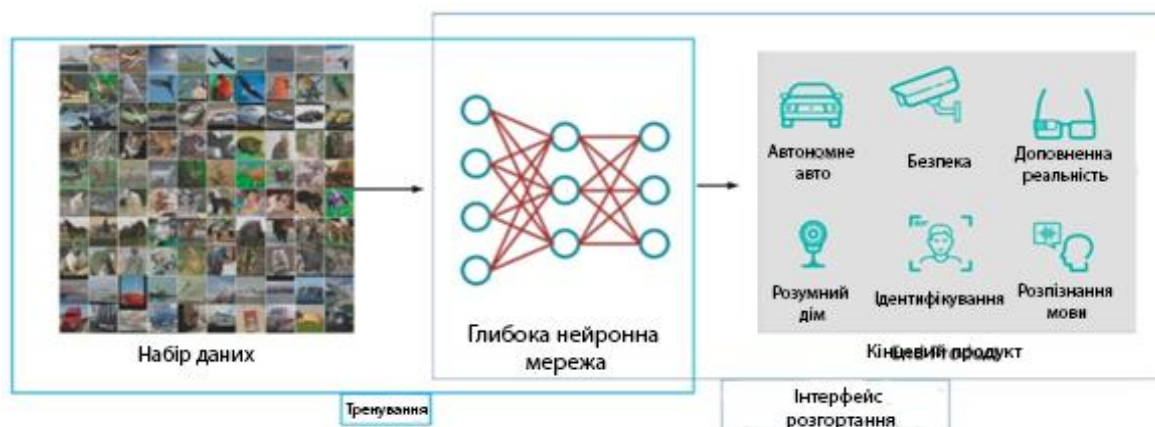


Рисунок 2.5 - Загальний потік розвитку глибокого навчання

Навчання - як тільки модель згорткової нейронної мережі буде розроблена, вона проходить навчання з відповідним набором даних. Висновок-навчена модель розгортається в кінцевому продукті і використовується для виведення з вхідними даними в режимі реального часу.

Метою роботи було досягнення наскрізного навчання, використовуючи лише зображення з камер як вхідний сигнал. Хоча для автономних

транспортних засобів використовується багато сенсорів, таких як лідар, радар, сонар, глобальна система позиціонування, інерційний блок вимірювання та одометрія коліс, камера є незамінним сенсором в автономному водінні, що дозволяє автономному транспортному засобу візуалізувати його оточення. Камери дуже ефективні при класифікації інтерпретації текстур, широко доступні і доступніші за інші сенсорів, що використовуються для виявлення, такі як радар або лідар. Обмеженням камери є обчислювальна потужність, необхідна для обробки даних.

Вхідними даними в нашій системі автономного водіння є лише зображення камери, пікселі. Вихід - це передбачення кута повороту рульового управління. Метою є досягнення обчислювальної моделі, яка може бути розгорнута на вбудованій автомобільній платформі для висновків у режимі реального часу. Розробити недороге рішення для машинного навчання з точки зору обчислювальної потужності та ресурсів пам'яті досягти непросто [58,59,60,61,62]. Методи, які забезпечують ефективну обробку глибоких нейронних мереж для поліпшення енергоефективності та пропускної здатності без шкоди для точності застосування чи збільшення апаратних витрат, мають вирішальне значення для широкого розгортання глибоких нейронних мереж у системах штучного інтелекту [62].

2.2.1 Автономна система водіння

У використаному підході використовувалось наскрізне навчання для автономної системи водіння. Вхідною інформацією в нашій системі автономного водіння було зображення (необроблені пікселі) а результатом – висновок щодо управління автомобілем. Було застосовано наскрізне навчання, завдяки якому мережа навчиться керувати транспортним засобом лише на основі вхідного сигналу з камери під час висновку в режимі реального часу.

Автономне керування в режимі реального часу - зображення, отримане з центральної камери, надходить на навчену модель глибокої нейронної мережі (DNN), а вихід цієї моделі - це прогнозування кута повороту.

Якщо використовувалося середовище тренажера для автономного водіння, транспортним засобом керував ручний (навчальний) режим водій - людина за допомогою клавіатури, миші або джойстика, і набір даних збирався автоматично. Дані, отримані в режимі ручного водіння, були зображеннями камери та значеннями кута повороту на кожен кадр. Зображення були використані як набір функцій, а вимірювання керма - як набір міток. Швидкість автомобіля було встановити легко, так як авто знаходиться в стимуляторі і можна з ідеальною точністю визначати швидкість. Дані, зібрані за допомогою цього підходу, були використані для навчання нейронної мережі, яка навчиться керувати автомобілем лише на основі вхідних даних, без будь-якої подальшої взаємодії з людьми. Цей метод також відомий як клонування поведінки. По-друге, глибока нейронна мережа для автономного водіння була навчена цьому набору даних для прогнозування кута повороту. Нарешті, навчена модель була використана для висновку, автономного водіння в режимі реального часу в середовищі симулятора. Показники успішного водіння на репрезентативній трасі полягали в тому, що автомобіль залишається на колії весь час під час автономного водіння.

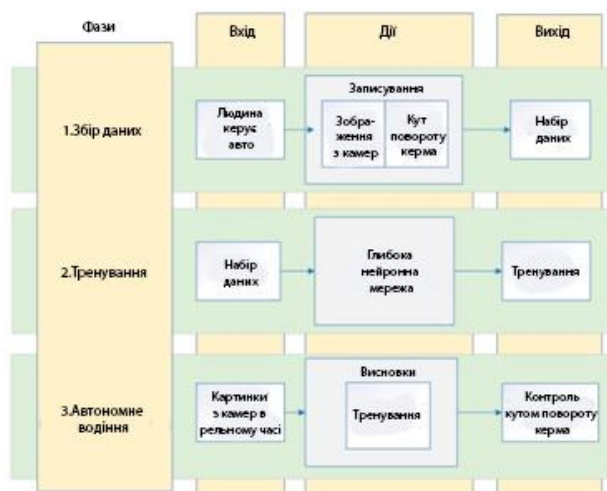


Рисунок 2.6 - Структурна схема: каркаса автономного водіння

Блок-схема автономної системи керування, яку було використано (див. Рисунок 2.6), представлена в показниках успішного водіння на репрезентативній трасі полягали в тому, що автомобіль залишається на колії весь час під час автономного водіння. Також в тому, що автомобіль залишається на колії весь час під час автономного водіння.

2.2.2 Симулятор середовища Unity

Платформою, яка використовувалась для збору даних та висновків, оцінки успішного автономного водіння, був симулятор автокерування автомобілем [7]. Автомобільний симулятор автомобіля вбудований у середовище розробки ігор Unity. Вид зверху на репрезентативну доріжку, що використовується для автономного водіння.

Для збору даних використовувалася репрезентативна доріжка для автономного водіння. Поки автомобіль їхав у ручному режимі, зображення з камер, встановлених у верхній частині автомобіля, записувалися разом із кутом повороту для цієї рамки. Дані з усіх трьох камер, встановлених зверху автомобіля, записували та зберігали разом із інформацією про кут повороту для одного кадру. Ця ж репрезентативна доріжка була використана для їзди в автономному режимі водіння, де рішення про кут повороту приймалося за допомогою зображення в реальному часі з камери, встановленої на транспортному засобі.

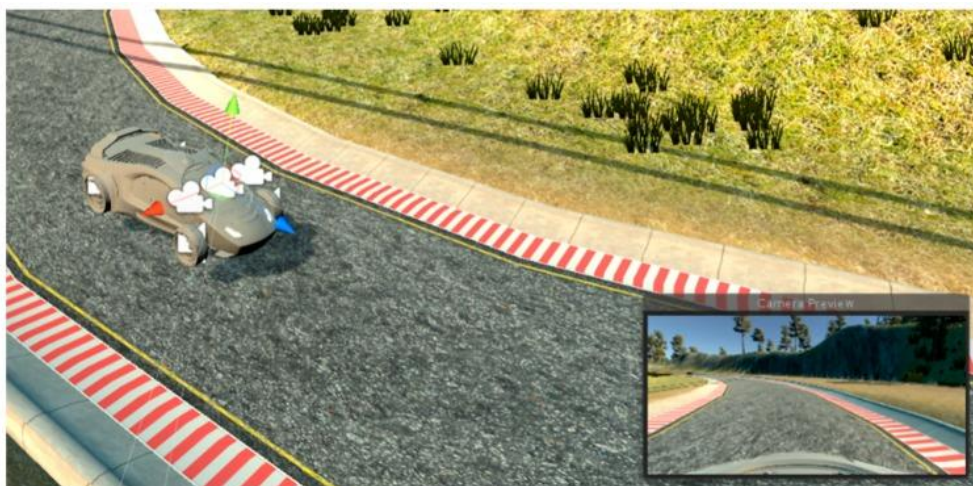


Рисунок 2.7 - Транспортний засіб в середовищі Unity

Транспортний засіб у тренажері на репрезентативній колії з трьома камерами зверху автомобіля(див. Рисунок 2.7). У нижньому правому куті - вид із центральної камери в цей конкретний момент.

Різні особливості представницької доріжки створюють проблеми для навчання наскрізної моделі автономного водіння. Наприклад, модель повинна навчитися поводитися з різкими поворотами, різною текстурою та різними межами дороги. Найскладнішими частинами репрезентативної траси для автономного водіння були три прямокутні криві одразу після мосту. Доріжка визначається червоно-білою смугою - брудом як кордон між доріжкою та рештою середовища тренажера. Приклади зображень, записаних центральною камерою в різних кадрах, які представляють різні характеристики дороги. Частина доріжки, захищена червоно-білою смугою, представлена у Рисунок 2.8 а). На вулиці можна побачити різну текстуру дороги Рисунок 2.87 d), що представляє міст на озері. На мосту дорога покрита цеглою, тоді як на інших частинах колії це переважно дорога з твердим покриттям. Більш того, кордоном цієї частини доріжки є невисока стіна. Частина доріжки, визначені брудом (з одного боку), та частини, визначені відбійниками, представлені в Рисунок 2.8 с) і Рисунок 2.8 d) відповідно. Ці різні особливості дороги в середовищі симулятора сприяють кращому узагальненню моделі, що призвело до успішного автономного водіння в різних сценаріях.



Рисунок 2.8 – Сценарії водіння

Вхідні дані, записані в різних кадрах за допомогою центральної камери. Під час представницької траси представлені різні характеристики дороги та її границь. Чотири основні межі колії та характеристики доріг такі: (а) червоно-

біла смуга - переважно різкі повороти; (b) невелика стіна - міст з різною текстурою дороги; (c) червоно-біла смуга з одного боку та бруд з іншого боку - криві; (d) відбійник - переважно пряма дорога.

Збір даних здійснювався під час руху автомобіля в ручному режимі по репрезентативній колії. Дані зображення були отримані з трьох камер, встановлених на транспортному засобі в середовищі імітатора. Наприкінці ручної їзди зображення зберігалися разом із таблицею, що містить інформацію про заголовки зображень та значення кута повороту для кожного записаного кадру. Приклад зображень, записаних усіма трьома камерами в одному кадрі. Для навчання використовували три камери. Під час збору даних у кожному кадрі зображення з трьох камер були зняті з однаковим значенням вимірювання керма. Незначна відмінність у полі зору для кожної центральної, лівої та правої камери призводить до кращого узагальнення моделі.

Приклад збору даних, знятих одночасно за допомогою трьох камер, встановлених на транспортному засобі: (a) ліворуч; (b) центр; (c) права камера. Це приклад зображень, зроблених у першому кадрі.

Для того, щоб мати якісні зразки, які дадуть змогу моделі засвоїти потрібні функції, транспортний засіб слід керувати в ручному режимі так само, як очікується, що він буде автономно їхати під час висновку. Під час збору даних мета полягала в тому, щоб транспортним засобом керували посеред дороги. Особливу увагу приділяли під час руху по прямокутних поворотах. Було дуже важливо, що наша модель навчилася правильно поводитися при поворотах.

Збільшення даних. Для збору даних було записано кілька кіл під час руху в ручному режимі, де збиралися дані з трьох камер. Навіть після кількох кіл зібраний набір даних був відносно невеликим. Тому застосовано методи збільшення даних. Однією з найважливіших особливостей, яку мала вивчити наша нейронна мережа, були криві. Простим методом подвоєння розміру набору даних та зосередження уваги на кривих було збільшення даних, коли

зображення переверталися у по горизонталі, а кут повороту помножувався на -1.

Загальна кількість отриманих зразків становила 34 288 зображень з роздільною здатністю $320 \times 160 \times 3$. Кожне записане зображення було 320 пікселів у висоту, 160 пікселів у ширину та три канали в глибину - кольорове зображення RGB. Середній розмір пам'яті одного записаного зображення становив близько 13,5 кБ. Кожне зображення було сполучене з відповідним значенням кута повороту, яке було нормалізовано в діапазоні від -1 до 1. Після застосування збільшення даних загальна кількість вибірок у наборі даних становила 68 576. Дані були розділені на категорії навчання та валідації, де 80% даних було обрано для навчання, 54 861 вибірок та 20% даних для валідації, 13 715 вибірок (Таблиця 2.1).

Тестування проводилося в режимі реального часу. Зображення в режимі реального часу з центральної камери безперервно надсилалися до навченої моделі машинного навчання, яка використовувалася для контролю кут повороту автомобіля. Оцінка успішної моделі проводилася шляхом спостереження, чи автомобіль міг їздити автономно протягом усієї заздалегідь визначеної колії. Якщо автомобіль був поза дорогою, це вважалось невдалим автономним керуванням.

2.2.3 Попередня обробка даних

Для навчання глибокої нейронної мережі використовувалися зображення, отримані з усіх трьох камер - центральної, лівої та правої. Використання трьох камер для збору даних для навчання глибокої нейронної мережі [1]. Усі ці зображення захопили схожу сцену, але з дещо різних позицій. Переваги використання трьох камер замість однієї центральної камери - це втричі більше даних і краща продуктивність для повернення до центрального сценарію, коли транспортний засіб починає збігати вбік. Кут рульового керування був поєднаний з трьома зображеннями з певного кадру, що відповідає центральній камері, а зображення з лівої та правої камер мали

поле зору, зміщене зліва чи справа від дороги, відповідно, що означає, що кермо кут для лівого та правого зображень неправильний. Щоб подолати це, коефіцієнт корекції для вимірювання кута повороту рульового колеса був доданий або віднятий відповідно до лівого та правого зображень. Корекційний коефіцієнт був одним із гіперпараметрів, які слід було доопрацювати під час навчального процесу.

Зображення з набору даних були нормалізовані шляхом поділу кожного пікселя зображення на 255, що є максимальним значенням пікселя зображення. Після того, як зображення було нормалізовано до діапазону від 0 до 1, було застосовано середнє центрування даних шляхом віднімання 0,5 на кожен піксель:

$$x_{norm} = x_{255} - 0,5 \quad (2.1)$$

що призвело до $-0,5 < x_{norm} < 0,5$.

Провідною ідеєю під час процесу проектування було досягнення наскрізного автономного водіння за допомогою легкої моделі (обчислювально найменш дорога модель), одночасно досягаючи найкращої продуктивності, у даному випадку автономного водіння на репрезентативному шляху. Найменш дорога в обчисленні модель - це зазвичай модель з найменшою кількістю параметрів, що впливає на їх пам'ять та обчислення. Тому тип і розмір шарів, розміри ядра та ряд карт функцій впливають на обчислювальну продуктивність. Працездатність автономного водіння перевіряли на симуляторі автомобіля, що керує автомобілем.

Остаточна архітектура моделі J-Net стала результатом експериментів із будівельними блоками глибоких нейронних мереж - різною кількістю шарів, розмірами ядра для згорткових шарів, кількістю карт функцій, розміщенням шарів об'єднання і, нарешті, експериментом з розміром і кількістю повністю

з'єднаних шарів. Структурна схема цього експериментального неглибокого CNN.

Першим кроком у розробці нового рішення стало використання надзвичайно дрібної CNN; Використання двовимірної операцію згортання над необробленими даними вхідного зображення:

$$S(i,j) = (Z * K.) (i,j) = \sum_v \sum_n Z(i-v_m, j-n) K.(v,n) \quad (2.2)$$

де триканальне вхідне зображення I мало розміри 320×160 , використовуваний двовимірний розмір ядра був 2×2 . Ядро використовувалося для вилучення латок зображення у згортковій операції. Результатом операції згортки був $S(i, j)$, двомірний тензор відображення ознак. Ваги w були розподілені між патчами для даного шару в CNN для виявлення конкретного зображення незалежно від того, де на зображенні воно розташоване. Рівняння для обчислення ширини та висоти згорткового вихідного шару є:

$$W_{out} = W - F + 2PS + 1 \quad H_{out} = H - F + 2PS + 1 \quad (2.3)$$

де W і H - ширина та висота вхідного шару, F - розмір фільтра (ядра), S - крок, P - заповнення, K - кількість фільтрів.

У першому експерименті вхідне зображення становило 320×160 , $F = 2$, крок 1, і використовувався відступ $P = 0$. Це призвело до розміру вихідних шарів: $W_{out} = 319$ і $H_{out} = 159$. Вихід глибина дорівнювала кількості фільтрів; $D_{out} = 16$ у цьому випадку. Обсяг виходу після згорткової операції становить:

$$Wout * Hout * Dout \quad (2.4)$$

що становило 811 536. Після згортки була застосована функція активації ReLU:

$$ReLU(x) = \max(0, x). \quad (2.5)$$

ReLU функція активації є найбільш часто використовуваною активацією в прихованих шарах DNN. Основною перевагою ReLU перед іншими функціями активації, такими як сигмоїдна або тан, є те, що з ReLU немає проблем із зникненням градієнта [59].

За цією згорткою слідував сплющений шар, де двовимірний шар був перетворений в вектор одновимірного розміру. Сплющений шар не приніс нових параметрів, які можна навчити, оскільки перетворення вузлів з попереднього шару в єдиний вимір. Нарешті, наприкінці, коли були просторові особливості зображення в результаті згортки, було застосовано повністю з'єднаний шар, де всі вузли з вирівняного шару об'єднані в єдиний вихід, який безпосередньо передбачив значення кутів повороту. Оскільки з мережі є єдиний вузол виведення, ця мережа – мережа регресії.

Модель була навчена з набором даних, раніше зібраним, з використанням вільної функції середньої квадратичної помилки (MSE) та оптимізатора Адама. Результати під час висновку в режимі реального часу, як і очікувалося, були поганими, автомобіль не зміг утримати курс на дорозі. Однак якісна оцінка ефективності показала, що модель все-таки засвоїла деякі корисні функції. Як видно з відео автономного водіння в середовищі тренажера [63], модель навчилася слідувати лінії, тож транспортний засіб їхав уздовж озера, слідуючи лінією між землею та водою. Цей невеликий експеримент показав, що обраний напрямок є правильним, але модель

потребує вилучення додаткових функцій для успішного автономного водіння.

Висновки цього експерименту були цінними для наступного етапу розробки:

1. Не використовувати весь розмір вхідного зображення. Частини зображення як небо або нижня частина зображення не мали значення для автономного водіння. Важлива дорога, криві та межі дороги, біло-червона смуга, бордюр, відбійник, міст.

2. Використовувати нормалізацію даних, щоб мати однаковий діапазон значень для кожного з входів моделі. Це може гарантувати стабільне зближення ваги та ухилів.

3. Використовувати операцію об'єднання, щоб зменшити кількість мережевих вузлів на наступному рівні, а отже, і кількість параметрів, що піддаються навчанню. Об'єднувальний шар зазвичай використовується для зменшення розміру виводу та запобігання його надмірному підбору.

4. Використовувати більше згорткових шарів, оскільки згортки відповідають за вилучення об'єктів. Перший експеримент показав, що згортковий шар може витягти деякі функції, необхідні для автономного водіння (наприклад, одна особливість - це лінія, за якою слід слідувати автономний транспортний засіб), але один згортковий шар був недостатньо хорошим для нашої мети, потрібно більше карт характеристик.

2.2.4 Архітектура J-Net

Введення більш прихованих шарів у глибоку нейронну мережу допомагає підвищити ефективність параметрів. Швидше за все, за допомогою чистих параметрів ви отримаєте набагато більшу продуктивність, заглибившись, а не ширше. На додаток до цього, глибокі нейронні мережі, що застосовуються до зображень, дуже ефективні, оскільки зображення, як правило, мають ієрархічну структуру, яку глибокі моделі вловлюють природним чином. Нижні шари глибоких нейронних мереж захоплюють

прості функції, такі як лінії або краї. Подальші шари вилучають складніші елементи, такі як геометричні фігури, а останні шари - це вилучення об'єктів. Оскільки метою нашої роботи було керувати транспортним засобом по репрезентативній колії, ознаки, які потрібно було вилучити, були не об'єктами, а просто простими рисами чи геометричними формами. З цієї причини, для нашої остаточної моделі,

З метою кращого вилучення функцій, вибрано три згорткові шари, щоб розширити вилучені об'єкти; Рівняння (2.2). Вибрано три згорткові шари з 16, 32 і 64 картами характеристик. На вході до першої згортки розмір становив $320 \times 65 \times 3$, після нормалізації та обрізання необробленого зображення застосовано розмір ядра 5×5 із 16 картами функцій. Виходячи з рівнянь (2.3) та (2.4), загальна кількість параметрів, що піддаються дресируванню, після першого згорткового шару становила 1216.

Оскільки мета досягти легкого (за розмірам) рішення, а згортка - це дуже об'ємна операція, яка додає значну кількість мережевих вузлів та ваги, присвоєні кожному з цих вузлів, потрібно було зменшити вибірку. Одним із варіантів вирішення цієї проблеми було б використання кроку під час згортки, щоб щоразу зміщувати фільтри на кілька пікселів і зменшувати розмір карти об'єктів. Однак це зменшення вибірки зображення може спричинити втрату деяких важливих функцій, оскільки воно видаляє багато інформації. Друге рішення щодо зменшення вибірки зображення - це операція об'єднання. Замість того, щоб пропускати одну в кожних двох згортках, було використано невеликий крок у поєднанні з усіма звивинами в околицях і об'єднували їх. Щоб зменшити розмір шарів глибокої нейронної мережі, застосовано операцію максимального об'єднання після кожного згорткового шару.

$$y = \max(X_i) \quad (2.6)$$

де X_i - значення однієї точки входу.

Першою перевагою використання максимального об'єднання є те, що ця операція не містить параметрів і не додає нових параметрів. Це зменшує ймовірність збільшення комплектації. По-друге, максимальне об'єднання часто дає більш точну модель. З іншого боку, оскільки згортки, які йшли нижче, працюють на нижньому кроці, обчислення моделі стає дорожчим. Крім того, введення нового шару в якості об'єднання додає для налаштування більше гіперпараметрів, таких як розмір області об'єднання та крок. Об'єднаний шар працює незалежно від кожного фрагмента глибини входу та змінює його розмір просторово. У нашій моделі обрано MaxPooling розміром 2×2 , який зменшує кожен зріз глибини на вході на 2, по ширині та висоті, відкидаючи 75%. Розмір глибини залишився незмінним, в цьому випадку.

Остаточне рішення для J-Net мало три згорткові шари, за кожним слідувала функція активації ReLU, описана в попередньому підрозділі, а потім розмір шару MaxPooling 2×2 . Застосування операцій згортки та MaxPooling призвело до 32 544 параметрів, що піддаються тренуванню, після останніх згортків. Оскільки розроблялося репрезентативну мережу навчання, ці вузли мали бути з'єднані з одним останнім вузлом. Після останнього згорткового шару було додано сплющений шар, щоб отримати одновимірний вектор параметрів. Сплющений шар не додавав нових параметрів, а скоріше, лише переставляв існуючі в одному вимірі. Нарешті, останні шари розробленого DNN були двома повністю з'єднаними шарами, перший з десятима вихідними вузлами, а другий лише з одним вихідним вузлом для прогнозування кута повороту. Остаточна кількість параметрів, що піддаються навчанню, для цієї архітектури та для вхідного розміру $320 \times 160 \times 3$ пікселів становила 150 965.

Впровадження. Щоб об'єктивно оцінити продуктивність нашої мережі, було повторно реалізовано три моделі нейронних мереж: LeNet-5 [13], AlexNet [6], і PilotNet [1], з невеликими змінами, щоб мати змогу сприяти наскрізному навчанню автономному водінню. Ідея полягала в об'єктивній оцінці нашої моделі J-Net у порівнянні з відомими мережевими

архітектурами. З цієї причини всі реалізовані моделі пройшли навчання за тим самим набором даних, які було створено, і перевірено на тренажері за тих самих умов. На жаль, автономне водіння за допомогою моделі LeNet-5 виявилось невдалим, транспортний засіб не зміг залишитися на колії, тому цю модель виключили з подальшого аналізу. Висновки за допомогою AlexNet та PilotNet були успішними в автономному водінні протягом усього треку.

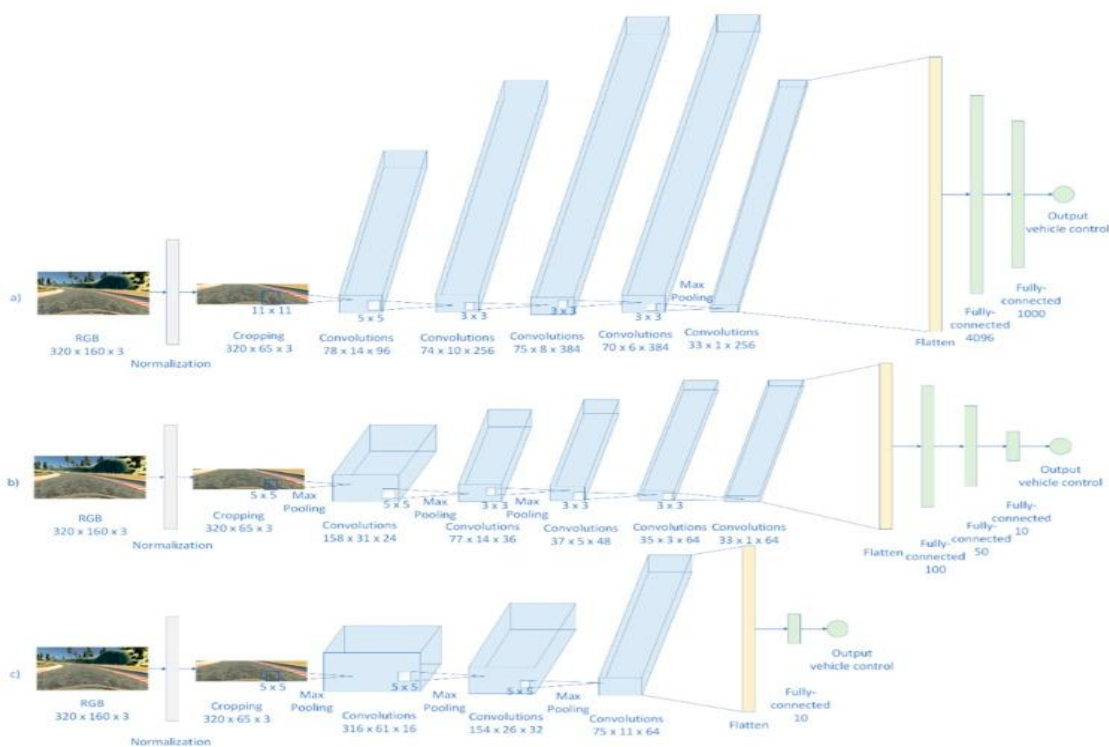


Рисунок 2.9 – Порівняння архітектур глибоких нейронних мереж

Оскільки архітектура J-Net описана в попередньому підрозділі, тут обговоримо деталі реалізації AlexNet та PilotNet. Порівняння мережевої архітектури J-Net, AlexNet та PilotNet показано на рисунку 2.11.

2.3 Реалізації AlexNet та PilotNet, порівняння з J-Net

У нашій роботі архітектура AlexNet [6] було повторно впроваджено та адаптовано з метою наскрізного навчання автономному водінню. В оригінальній архітектурі AlexNet є два паралельні конвеєри обробки. Це з

історичних причин, оригінальний AlexNet був навчений з використанням двох графічних процесорів (GPU), і це є причиною того, що вони розділили згортки на дві паралельні гілки. Оскільки було достатньо апаратних ресурсів для навчання AlexNet за допомогою одного графічного процесора, у нашій реалізації AlexNet було спрощено архітектуру та використали один потік згортки, використовуючи комбіновану кількість фільтрів, запропонованих вихідним рішенням. Інші відмінності між оригінальною архітектурою та нашою реалізацією цієї архітектури полягали у розмірі вхідної інформації, для AlexNet це було 224×224 пікселів, для нашого випадку вхідне зображення - 320×160 пікселів, і після обрізання, яке було потрібно для відхилення непотрібної інформації, що міститься в крайових пікселях, розмір зображення, з яким мали справу наші реалізації, становив 320×65 пікселів. Різниця у вхідних шарах вплинула на це у втіленні AlexNet, тому пропущено два шари об'єднання. Введення всіх трьох шарів об'єднання значно зменшило б кількість пікселів, доступних для наступних згортків, а оскільки було менше вхідного зображення в одному вимірі, не вистачило б пікселів для всіх згорткових шарів із використанням оригінальних ядер. Для того, щоб спростити та змінити мережу для наших цілей, було випущено перші два шари об'єднання з вхідної мережі та мали один максимальний шар об'єднання після четвертого згорткового шару з 384 функціями та до останнього згорткового шару з 256 функціями, що було потрібно для відхилення непотрібної інформації, що міститься в крайових пікселях, розмір зображення, з яким мали справу наші реалізації, становив 320×65 пікселів.

Різниця у вхідних зображеннях між оригінальною архітектурою та нашою повторною реалізацією вплинула на загальну кількість параметрів. В даному випадку була менша кількість параметрів, що піддаються навчанню, ніж це було в оригінальній архітектурі. Оригінальна версія AlexNet містить понад 63 мільйони параметрів, що піддаються навчанню, тоді як наша повторна реалізація AlexNet мала понад 42 мільйони параметрів, що піддаються навчанню. Остання відмінність між оригінальною архітектурою

та нашою реалізацією AlexNet - це останній шар. В оригінальній версії AlexNet в якості вихідного шару було 1000 вузлів, оскільки на конкурсі ImageNet було 1000 класів. В даному додатку був потрібен лише один вузол виводу: передбачення кута повороту рульового управління для автомобіля.

PilotNet, також званий NVIDIA CNN, розроблений дослідниками з корпорації NVIDIA,-це глибока нейронна мережа для наскрізного автономного водіння, створена з метою вдосконалення системи автономного водіння DARPA (DAVE) для автономного водіння [1,2]. PilotNet має рівень нормалізації, за яким слідує п'ять згорткових шарів, за якими йдуть чотири повністю з'єднані шари. Різниця між оригінальною архітектурою PilotNet та нашою реалізацією PilotNet - це введення даних мережі, що призвело до невеликої різниці у розмірах шарів та кількості параметрів, що піддаються навчанню. У оригінальній PilotNet площа введення становить 200×66 пікселів, тоді як в даному випадку введення становило 320×160 пікселів, а після обрізання - 320×65 пікселів. Порядок решти згорткових, сплюснених і повністю з'єднаних шарів був подібним, з тією ж кількістю карт характеристик для згорткових шарів. Оскільки розмір вихідного входу в PilotNet був не таким, як розмір нашого входу, була різниця в кількості параметрів, які можна навчити. Наприклад, сплюснений шар після п'ятого згорткового шару, в оригінальній архітектурі PilotNet налічується 1164 нейрони, тоді як у нашій реалізації PilotNet було 2112 нейронів. Крім цього, наша повторна реалізація PilotNet йшла за оригінальною моделлю.

2.3.1 Стратегія навчання та настройка

Три архітектури J-Net, AlexNet та PilotNet були реалізовані в Python за допомогою Keras [64], бібліотека глибоких нейронних мереж високого рівня, написана поверх бібліотеки TensorFlow. Було впроваджено та навчено всі мережі з нуля. Платформою, яка використовувалася для навчання та взаємодії, був ПК з 12-ядерним процесором на 3,2 ГГц та графічний процесор NVIDIA GeForce GTX 1070-Ti з 8 ГБ GDDR5 та 8 Гбіт /с швидкість пам'яті.

Під час навчання зображення лівої та правої камери надходили до моделі так, ніби вони надходили з центральної камери. Використання трьох камер втричі збільшило зібрані дані і допомогло моделі навчитися керувати автомобілем, якщо він злітає ліворуч або праворуч. Зображення з центральної камери були зроблені без зарядів, тоді як для зображень з лівої та правої камер параметр корекції застосовувався до вимірювань кута повороту. Після точної настройки було обрано коефіцієнт корекції 0,22, який потрібно додати або відняти відповідно з лівого та правого зображень відповідно. На додаток до цього, попередня обробка даних, яка була застосована тут, включала обрізання зображень, нормалізацію даних та середнє центрування даних.

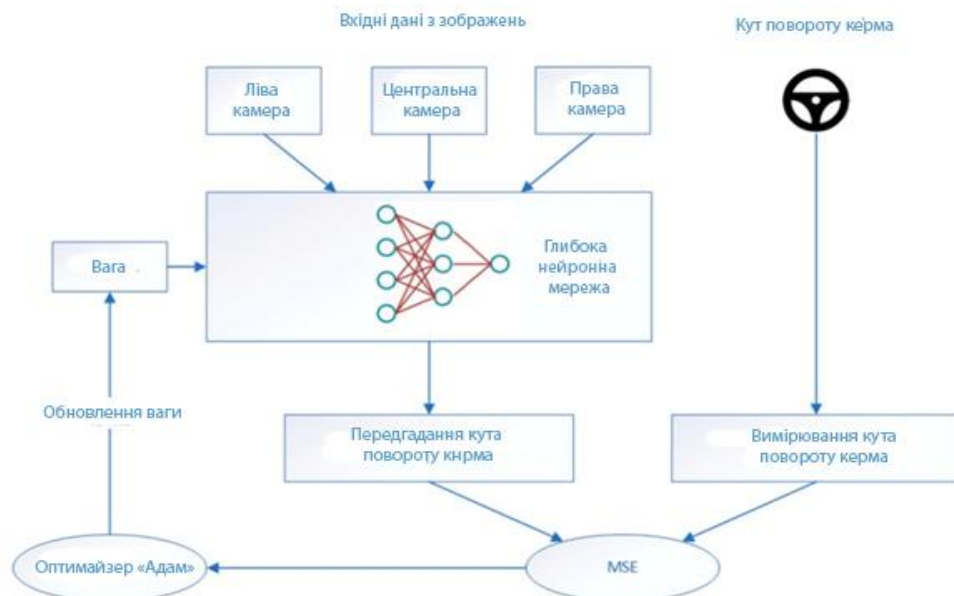


Рисунок 2.10 – Навчальна схема

Збір даних здійснюється в ручному режимі водіння шляхом отримання зображень із трьох камер, встановлених на транспортному засобі: центральної, лівої та правої паралельно із записуванням кута повороту в парі з кожним зображенням. Під час навчального процесу до лівого та правого зображень додано коефіцієнт корекції.

Кожну з розроблених моделей навчали окремо, використовуючи раніше зібраний набір даних. Для функції втрат була використана середня квадратична помилка, щоб мінімізувати похибку між передбаченням рульового керування та землею шляхом вимірювань рульового управління. Було обрано MSE, оскільки це є відповідною функцією втрат для регресійних мереж [8,9,10].

Оскільки було використано порівняно невеликий набір даних, існувала ймовірність того, що станеться похибка. Тому був застосований метод ранньої зупинки регуляризації. Розроблені моделі глибоких нейронних мереж J-Net, AlexNet та PilotNet навчалися з різною кількістю епох. Спочатку моделі навчалися за 30 епох. Однак лише для AlexNet ця кількість епох дала хороший результат; втрати для валідації зменшувалися в той же час, що і втрати для навчання. AlexNet показав хороші результати в автономному водінні, повністю завершивши завдання наскрізного автономного занурення на тренажері. З іншого боку, навчання PilotNet та J-Net за 30 епох показало переобладнання, втрати від перевірки почали зростати, а втрати від навчання продовжували падати. Використовуючи цю кількість епох для навчання PilotNet та J-Net, автомобіль успішно їхав автономно на одній частині дороги, але після того, як автомобіль з'їхав з повороту, автомобіль з'їхав з дороги. Для того, щоб запобігти переналадженню, по-перше, спробуємо метод відсіву, застосований після першого щільного шару в мережевих архітектурах.

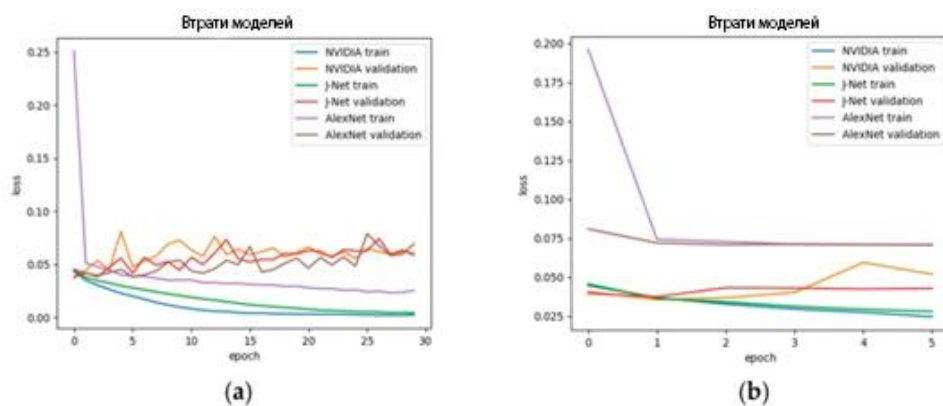


Рисунок 2.11 – Результати навчання моделей

Втрата моделі для навчання та валідації для AlexNet, PilotNet (тобто NVIDIA CNN — згортоква нейронна мережа корпорації NVIDIA) та J-Net (a) для навчання протягом 30 епох; (b) для навчання у 6 епохах (див. Рисунок 2.9).

В кінцевому рішенні був застосований метод регуляризації з ранньою зупинкою. Як і очікувалося, дострокове припинення забезпечило менші значення втрат валідації для моделей J-Net та PilotNet, тоді як значення втрат валідації для AlexNet залишилося на аналогічному рівні. Оскільки використовуваний набір даних для навчання був однаковим для всіх моделей, різниця у співвідношенні між тестуванням та втратами валідації на модель, навчену з різною кількістю епох, була безпосередньо пов'язана з самою архітектурою мережі. І моделі J-Net, і PilotNet мали меншу кількість параметрів, що піддаються навчання, ніж AlexNet. Тому для одного і того ж розміру набору даних було більш ймовірно, що для цих двох моделей відбудеться переобладнання. Після того, як був застосований метод раннього припинення регуляризації для переобладнання, моделі J-Net та PilotNet мали очікувану реакцію, зменшення надмірності.

У PilotNet в обох випадках - 30 епох та 6 епох - це видно з a, b(див . Рисунок 2.11) про те, що в п'яту епоху був пік втрати валідації, помітна різниця між валідацією та втратою навчання, що вказувала на необхідність застосування методу раннього припинення та вибору чотирьох епох для навчання PilotNet. Валідація моделі під час автономного водіння підтвердила цей висновок, показавши, що PilotNet забезпечив найкращі показники водіння, коли її навчали чотири епохи. З іншого боку, вибір шести епох для навчання моделі J-Net був більш емпіричним вибором. Для навчання моделі J-Net експерименти показали, що втрати валідації були отримані з меншою кількістю епох, але це число могло коливатися від чотирьох до 10 епох з подібними результатами. Було обрано шість епох, які забезпечили успішне автономне водіння під час валідації моделі.

Навчена модель була збережена і використана пізніше для висновку про автономне водіння. Як і очікувалося, виходячи із загальної кількості параметрів, що піддаються навчанню, описаних у попередньому розділі, найлегшою моделлю була J-Net, яка мала всього 1,8 МБ. Модель, яка потребувала найбільшого простору пам'яті, була AlexNet з 509,5 МБ, яка відповідає її кількістю параметрів, що піддаються дресируванню непідготовленої мережі, понад 44 мільйонам параметрів. Навчена модель PilotNet мала об'єм пам'яті 4,2 МБ.

Запропоновану глибоку нейронну мережу J-Net порівняли з AlexNet та PilotNet, які було повторно реалізовано, щоб провести об'єктивну оцінку ефективності нового дизайну. Моделі всіх трьох мережевих архітектур були реалізовані, пройшли навчання з тим самим набором даних, а навчені моделі були використані для висновку в тренажері для автономного водіння. Результати порівнювались з точки зору продуктивності, успішного руху на репрезентативній трасі, а також з точки зору складності мережі, ряду параметрів, що піддаються навчанню, та розміру навчальної моделі.

2.3.2 Обчислювальна складність

Виконання моделей глибоких нейронних мереж сильно залежить від набору статичних констант, ваг, які також називаються параметрами моделі. Сама архітектура мережі, зв'язки між вузлами, безпосередньо визначає обчислювальну вартість мережі. Однією з основних відмінностей між звичайними нейронними мережами та звичайними штучними нейронними мережами є те, що зв'язки між нейронами не повністю зв'язані. Таким чином, особлива організація глибокої нейронної мережі та точна характеристика обчислень у фільтруючих елементах визначають складність мережі.

Для того, щоб кількісно виміряти обчислювальний попит кожної навченої мережі, складність мережі, порівняли кількість параметрів, що піддаються навчанню. Глибина мережі, кількість шарів та типи шарів, згортки, об'єднання та щільність однозначно визначають кількість параметрів

мережі. Як видно з таблиці, у AlexNet було 42 452 305 параметрів, що піддаються навчанню, у PilotNet-328 219 параметрів, що піддаються навчанню, тоді як у J-Net було лише 150 965 параметрів, що піддаються навчанню. Представлена в цьому документі мережа J-Net мала приблизно половину параметрів, що піддаються навчанню, ніж PilotNet, і приблизно в 280 разів менше, ніж AlexNet.

Таблиця 2.1- Шари та кількість параметрів для навчання для J-Net, PilotNet та AlexNet.

Шари та параметри	AlexNet	PilotNet	J-Net
Точність	5	5	3
Вирівнювання	1	1	1
Щільність (повністю підключений)	3	4	2
Загальна кількість параметрів, що навчаються	42 452 305	348 219	150 965
Операції множення та доповнення	42,45	347,82	150,84

Крім того, порівняємо розмір навчених моделей: модель AlexNet мала об'єм пам'яті 509,5 МБ, PilotNet 4,2 МБ, а J-Net-лише 1,8 МБ;Таблиця 2.2. Усі моделі навчалися з однаковим набором даних, функцією втрат та оптимізатором. Кількість епох, які використовувалися для навчання кожної моделі, була різною через відмінності у надмірній комплектації моделі, на що вказує співвідношення тренувань та втрат валідації, отриманих під час навчання моделі. Хоча сама мережева архітектура, тип шарів мережі та

зв'язки між шарами безпосередньо впливають на обчислювальну вартість, розмір навченої моделі впливає на умовиводи через обмеження пам'яті вбудованих апаратних платформ.

Таблиця 2.2 - Розмір навчальних моделей: J-Net, PilotNet та AlexNet.

	AlexNet	PilotNet	J-Net
Кількість епох, використаних для навчання	30	4	6
Розмір навченої моделі	509,5 МБ	4,2 МБ	1,8 МБ

Нова мережа, яку було запропоновано, J-Net, мала близько 150 тисяч параметрів для навчання, що складало половину нашої реалізації PilotNet, яка мала близько 350 тисяч параметрів для навчання, а модель J-Net мала параметри в 280 разів менше, ніж повторна реалізація AlexNet, який мав понад 41 мільйон параметрів, що піддаються навчанню, а це означає, що нам вдалося створити найменш обчислювальне рішення. Розмір моделі навчання J-Net був у чотири рази меншим за модель PilotNet і приблизно у 250 разів менший за модель AlexNet. Менший розмір мережі та кількість параметрів, що піддаються навчанню, призвели до покращення продуктивності в режимі реального часу з точки зору скорочення затримок, а також до зменшення потреби у взаємодії обладнання з точки зору обчислювальної потужності, вартості та простору.

Виходячи з цих результатів, можна сказати, що запропонована нами мережа мала менш глибоку архітектуру, ніж інші рішення, з якими її порівнювали, меншу кількість параметрів, що піддаються навчанню, і, отже, була меншою навченою моделлю. Це рекомендує розроблену мережу для розгортання на вбудованих автомобільних платформах

2.3.3 Результати перевірки в тренажері

Перевірка автономного водіння проводилася в тренажері на представницькій доріжці. У режимі автономного водіння сигнал від центральної камери, встановленої на транспортному засобі, безперервно отримувався та надсилався як вхід до навченої моделі машинного навчання, що призвело до контролю кута повороту. Автономне водіння з використанням усіх трьох моделей було записано та наведено у відеороликах у [66,67,68] для AlexNet, PilotNet та J-Net відповідно. Як видно, J-Net виконала вимогу щодо автономного водіння за заздалегідь визначеним маршрутом, коли автомобіль залишався на дорозі протягом усього часу їзди. Вимірювання продуктивності було успішним проїздом на репрезентативній трасі, поведінкою, що автомобіль не зійшов з колії під час їзди, що означає, що найкращим рішенням було те, де транспортний засіб знаходився посередині колії протягом повної тривалості поїздки. Продуктивність моделі J-Net була задовільною. Якісна оцінка ефективності автономного водіння за допомогою впроваджених мереж наведена в Таблиця 2.3.

Таблиця 2.3 - Якісна оцінка продуктивності автономного водіння за допомогою AlexNet, PilotNet та J-Net.

Автономне водіння	AlexNet	PilotNet	J-Net
Автономна їзда по репрезентативній трасі	Успішно	Успішно	Успішно
Обробка кривих	Добре	Середнє	Добре
Дотримуючись центру траєкторії	Добре	Середнє	Середнє
Водіння по різних текстурах поверхні	Добре	Добре	Добре

Одним із показників оцінки продуктивності було спостереження за поведінкою транспортного засобу на кривих. Серед цих трьох рішень AlexNet показав найкращі результати під час автономного водіння. Використовуючи AlexNet для автономного водіння, більшу частину часу автомобіль перебував посеред дороги, тоді як під час автономного водіння за допомогою PilotNet та J-Net автомобіль майже завжди знаходився посеред дороги, але на деяких поворотах він наближався до краю. Однак усі три реалізації автономного водіння завжди мали змогу керувати транспортним засобом на дорозі, і не зійшли зі шляху.

Окрім спостереження за автономним водінням на репрезентативній трасі, були оцінені прогнози кута повороту, що використовуються для автономного водіння. Як видно, прогнози кута повороту для всіх трьох моделей були відносно подібними. Графічне представлення прогнозів кутів повороту, що використовуються для висновків у реальному часі, наводиться для одного повного кола автономного водіння на репрезентативній трасі. Позитивні та негативні значення кута повороту позначають лівий і правий кут повороту. Оскільки репрезентативна доріжка, яка використовувалася для їзди під час висновку, була однаковою, а оскільки швидкість автомобіля була виправлена через простоту, кут повороту показує прогнози кутів повороту в подібних рамках. Прогнози кута повороту для моделі J-Net та PilotNet були подібними; однак, J-Net мав дещо вищі значення в обох напрямках, ліворуч і праворуч. Модель AlexNet дала переважно плавні прогнози рульового управління в більшості поїздок. Однак у деяких точках він мав екстремальні значення, наприклад, приблизно на 2500 кадрах спостерігається спайк у лівому напрямку, тоді як інші дві моделі не мали такого різкого повороту для цієї частини дороги.

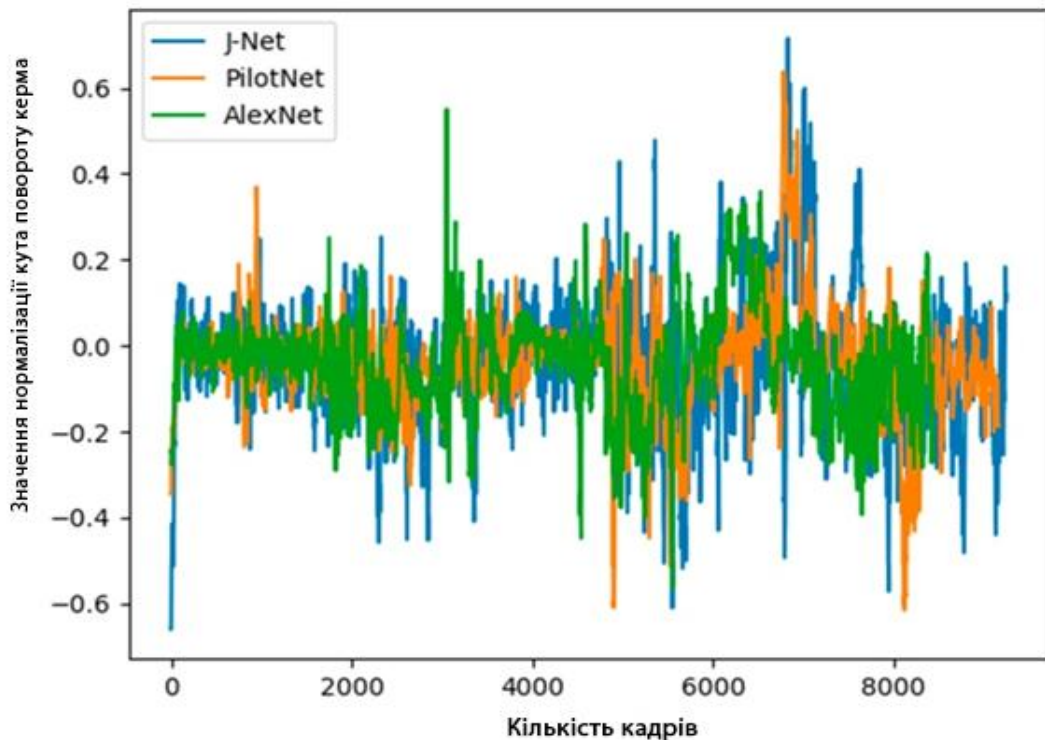


Рисунок 2.12 – Передбачення кута повороту руля

Передбачення кута повороту руля, що використовується для автономного водіння, представлені як нормовані абсолютні значення кута повороту в градусах. J-Net має синій ділянку, PilotNet помаранчевий і AlexNet зелений (див. Рисунок 2.12).

Як видно з Рисунок 2.11, водіння з використанням усіх трьох моделей мало подібні моделі. Моделі показують, що автомобіль їде переважно без коливань на прямих ділянках дороги. Однак на кривих відхилення від центру траєкторії було найбільшим (наприклад, після моста - частина діаграми в Рисунок 2.11 позначені як (с), є три гострі криві - Рисунок 2.11 d, b, тоді як третя крива була найскладнішою). Крім того, Рисунок 2.11 показує, що всі три мережі мали відхилення в цій частині, де AlexNet мав найбільше відхилення, а J-Net працював краще, ніж інші моделі. З іншого боку, J-Net мав більше коливань протягом повного кола, тоді як AlexNet мав найкращі показники, будучи найближчим до центру траєкторії протягом більшої частини їзди.

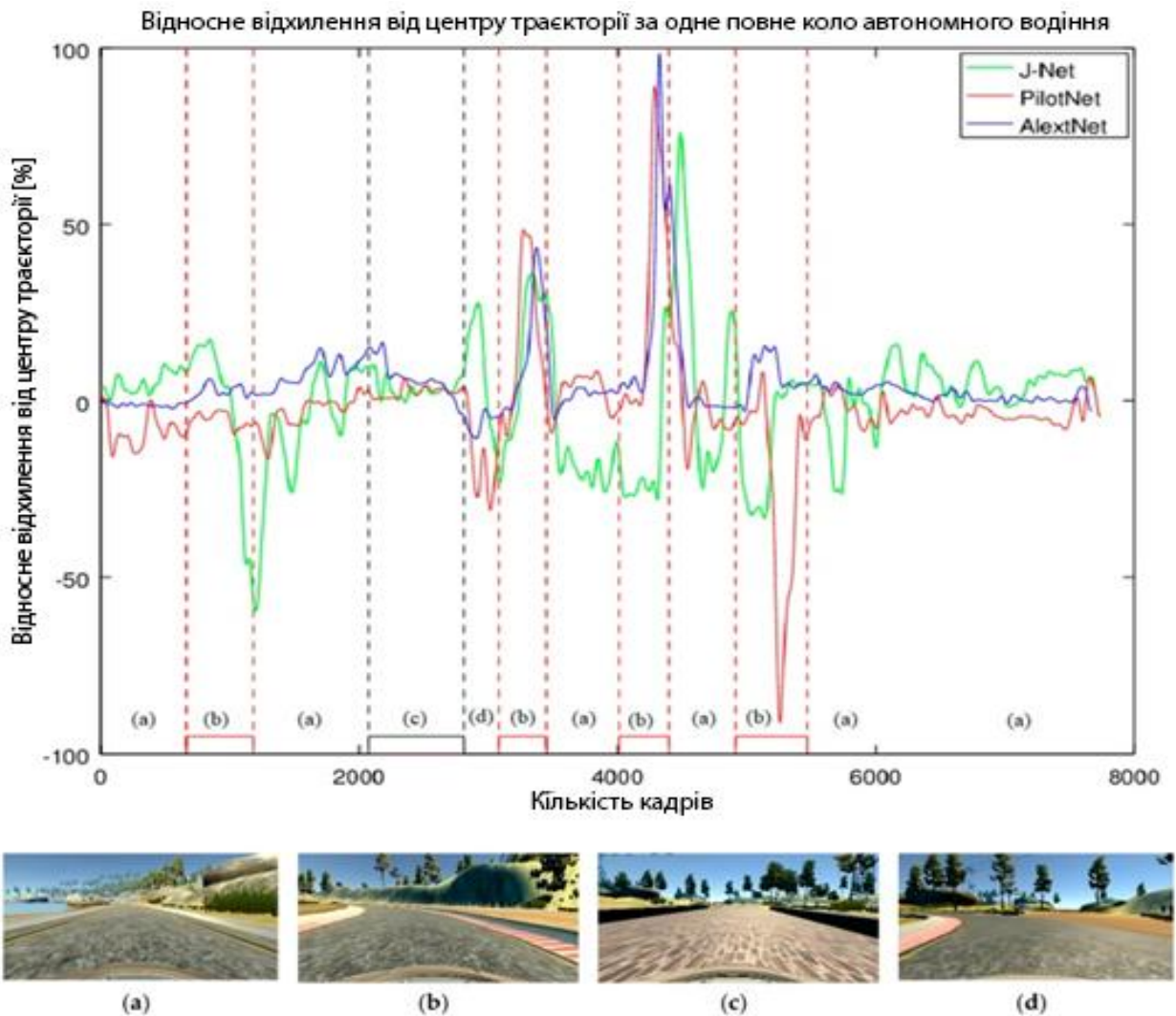


Рисунок 2.13 – Результати тестування при різних умовах

Відносне відхилення від центру траєкторії за один повний круг автономного водіння. Чотири основні характеристики траєкторії - це частини колії, визначені: (a) відбійниками - звичайна, переважно пряма дорога; (b) червоно-біла смуга - переважно гострі криві; c) невелика стіна - міст; (d) червоно-біла смуга та бруд - гостра крива (див. Рисунок 2.13).

Статистичний аналіз автономного водіння також представлений за допомогою гістограм. Цей аналіз є важливим для довгострокових випробувань. Для того, щоб дослідити коливання, гістограма відносних відхилень від центру траєкторії за один повний круг автономного руху представлена в Рисунок 2.13. Гістограма водіння J-Net представлена в Рисунок 2.13 а), де було показано, що J-Net має найменше відхилення в

кривих, тоді як коливання для центру траєкторії були найбільшими. Для пілотної мережі, Рисунок 2.13 b), коливання були середніми в порівнянні з використанням двох інших мереж для автономної їзди, але ця модель мала значні коливання в кривих, як ліворуч, так і праворуч, як видно з Рисунок 2.13 b), де для обох напрямків спорадичні випадки мали відносне відхилення від центру траєкторії майже на 100%. Гістограма відносного відхилення від центру траєкторії показує, що використання AlexNet для автономного водіння, мав найстійкіший досвід водіння з найменшими коливаннями від центру траєкторії. З іншого боку, спостерігалось спорадичне велике відхилення від центру траєкторії в одному напрямку кривої. Однак це відхилення було в дозволених межах, машина не вийшла з дороги, що було критеріями, які визначено для успішного автономного водіння.

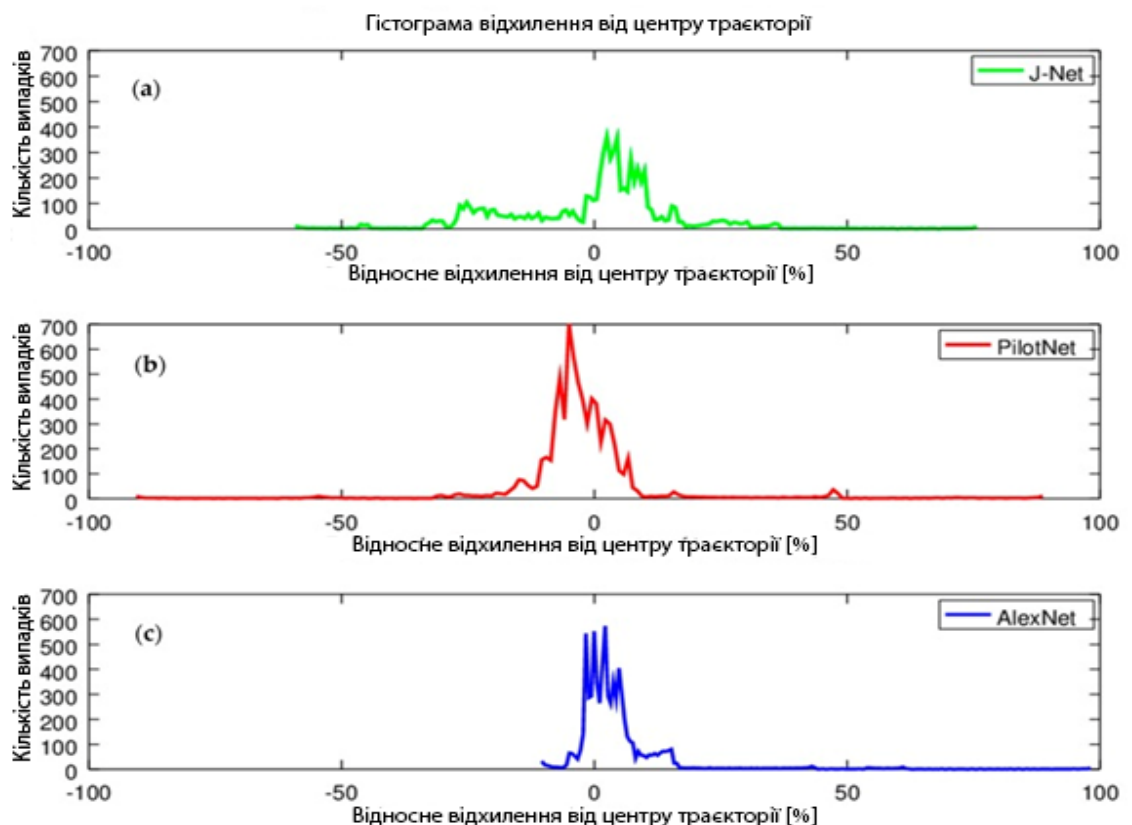


Рисунок 2.14 – Гістограми відхилення від центру траєкторії за один повний круг автономного водіння. (a) J-Net; (b) Pilot-Net; (c) AlexNet.

Нарешті, всі моделі показали себе добре, успішно завершивши круг автономного водіння без значних відхилень від центру траєкторії. Відмінності між автономним водінням на різних моделях були помітними, але не великими.

На основі аналізу складності обчислень очікувалося, що J-Net матиме найменшу затримку та найвищу частоту кадрів серед трьох оцінених рішень. Кількісна оцінка ефективності підтвердила це твердження, як видно з Таблиці 2.4. Ця оцінка була зроблена на платформі ПК, яка є високопродуктивною платформою, яка використовується для середовища симулятора. J-Net змогла успішно виконати завдання автономного водіння на представницькій трасі. Оскільки траса є замкнутою, було виміряно кількість успішних послідовних кіл усього за десять кіл. Усі три моделі змогли успішно керувати автомобілем протягом вимірюваного часу. Для вимірювання затримки було розраховано час між двома послідовними прогнозами. Це число змінюється протягом всього кола автономного водіння, тому для затримки використовували середнє значення. Кадри в секунду обчислювали шляхом підрахунку кількості передбачень на отримані кадри за одну секунду.

Таблиця 2.4 - Кількісна оцінка продуктивності автономного водіння за допомогою AlexNet, PilotNet та J-Net на високопродуктивній платформі, що використовується для середовища тренажера.

	AlexNet	PilotNet	J-Net
Кількість успішних кіл	10	10	10
Затримка	26,0мс	24,1мс	23,8мс
Кількість кадрів в секунду	37 кадр / с	42 кадр / с	44 кадр / с

Автономне керування випробовується на 10 колах поспіль. Спостереження за вимірюванням частоти кадрів за допомогою моделі J-Net для виведення в реальному часі було на 30% швидшим, ніж при використанні моделі AlexNet на високопродуктивній платформі, що використовується для середовища імітатора. Однак, якби я використовував скалярний процесор для висновку, можна було б очікувати значних відмінностей (наприклад, використання J-Net через AlexNet було б у 280 разів швидше). В експерименті, де було використано середовище симулятора, платформою зараження був комп'ютер великої ємності з графічним процесором, який забезпечував паралелізацію даних. Отже, ці результати були для конкретного застосування, де використовувався графічний процесор. Тут, оскільки архітектури нейронних мереж відрізнялися більшою площею поверхні, ніж глибиною, більшість операцій можна було виконувати паралельно,

Правильна демонстрація переваг продуктивності J-Net залежить від платформи. Якщо прийдемо до іншої крайності, коли операції виконуються тільки на скалярних процесорах, очікується, що швидкість виконання буде значно відрізнятися, тобто порівняти з ємністю мережі, кількістю параметрів. Реальні реалізації моделі J-Net призначені для вбудованих платформ, в яких ступінь паралелізації буде встановлено таким чином, щоб виконувалися вимоги до продуктивності в частоті кадрів.

2.4 Висновки до розділу 2

Підводячи підсумок, це був дійсно цікавий і водночас складний проект для роботи. Глибоке навчання - це захоплююча сфера, і нам пощастило жити в один час з такими можливостями.

У другому розділі було розроблено проект для автономного водіння, провели масштабний тест трьох моделей безпілотного авто в стимуляторі водіння (віртуальному середовищі), в якому оцінили переваги та недоліки різних підходів до моделей безпілотного автомобіля. Пройшли шлях від ідеї

до реалізації, оцінили всі можливі види сенсорів, їх злиття. За допомогою симуляції різних середовищ провели тест поколінь нашої нейронної мережі і змогли навчити машину керувати собою. Підвели фінальну статистику. І незалежно оцінили всі вихідні дані.

Зважаючи на стрімкий ріст технологій автономного водіння, через 10 років більшість із нас, ймовірно, не матимуть автомобіля. Буде якась підписка в якійсь компанії, наприклад Uber, і потрібно буде платити 149 доларів на місяць, і кожного ранку людина буде прокидтися з автомобілем біля під'їзду, який доставить її на роботу.

3 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

3.1. Організація охорони праці на виробництві автомобільного транспорту

В даній роботі розглядається розробка і навчання моделі безпілотного авто за допомогою нейронної мережі, там було використано віртуальний простір для цього. Науково-технічний прогрес вносить принципові нововведення у всі сфери сучасного матеріального виробництва, при цьому докорінним чином змінюються знаряддя та предмети праці, методи обробки інформації. Актуальна темою є проблема захищеності працюючого населення; організації охорони праці на підприємстві, в установі, організації, створенні безпечних умов праці.

Охорона праці – це система правових, соціальних, організаційних, технічних, санітарно-гігієнічних та лікувально-профілактичних засобів і заходів, які спрямовані на забезпечення безпеки життя та здоров'я робітників під час трудової діяльності [69].

Якщо кількість працюючих 50 або більше, на підприємстві повинна бути створена служба охорони праці, а також розроблено «Положення про службу охорони праці». На виробництвах де працівників менше 50, роботу служби охорони праці можна виконувати в порядку сумісництва. Якщо на виробництві менше 20 людей то роль служби охорони праці може виконувати сторонній фахівець, який має стаж не менше трьох років і успішно пройшли навчання з охорони праці [70].

Організація охорони праці на підприємстві покладається на роботодавця. Останній зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав

працівників у галузі охорони праці. З цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

- створює відповідні служби і призначає посадових осіб, відповідальних за вирішення конкретних питань охорони праці, затверджує інструкції про їх обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх додержання;
- розробляє за участю сторін колективного договору і реалізує комплексні заходи для досягнення встановлених нормативів та підвищення існуючого рівня охорони праці;
- забезпечує виконання необхідних профілактичних заходів відповідно до обставин;

Інструктаж з питань охорони праці проводиться всім новим працівникам, також, під час інструктажу, працівника повинні повідомити про умови праці. До умов праці включаються всі небезпечні та шкідливі виробничі фактори, можливі наслідки для здоров'я під час роботи. Крім того, обов'язковим є проходження перевірки знань.

Машинобудування - це робота з підвищеним ризиком. Тому, в випадку автомобільного виробництва, працівники повинні щорічно проходити додаткове навчання і перевірку знань [69]. Працівники під час прийняття на роботу проходять інструктаж та навчання охороні праці, безпеці, правилам пожежної поведінки, правилам надання домедичної допомоги, правилам поведінки під час аварій. Все це проводиться за рахунок роботодавця.

Роботодавець зобов'язаний за свої кошти забезпечити фінансування та організувати проведення попереднього (під час прийняття на роботу) і періодичних (протягом трудової діяльності) медичних оглядів працівників, зайнятих на важких роботах, роботах із шкідливими чи небезпечними умовами праці або таких, де є потреба у професійному доборі, щорічного обов'язкового медичного огляду осіб віком до 21 року.

Якщо цього потребує місце роботи працівникам повинні бути виданні засоби індивідуального захисту [71]. Це стосується робіт пов'язаних з небезпечними умовами, забрудненнями, несприятливими температурними умовами. Також це стосується специфічних роботи які потребують додаткових засобів (рідни для очищення, додаткових засобів гігієни тощо). Всі трати на індивідуальних захист бере на себе роботодавець [69], якщо працівник купує ЗІЗ сам, то роботодавець зобов'язаний компенсувати вартість, але лише в тому випадку, якщо роботодавець вчасно не зміг надати ЗІЗ працівнику.

Атестація робочого місця є важливим аспектом, особливо якщо потрібно побудувати підприємство для виготовлення автомобілів. Це зумовлено великою кількістю обладнання, металів, сировини, всі ці фактори впливають на безпеку працівників. Періодична атестація, допомагає уникнути нехтування правилами безпеки та безвідповідальному відношенню до обладнання, сировини, робочого місця та правил безпеки. Така атестації повинна проводитися спеціально підготовленою комісією і не менше ніж один раз в 5 років.

Якщо трапився нещасний випадок, роботодавець має організувати розслідування, для того, щоб встановити причини. Результатом розслідування має бути визнаний винуватець (це може бути сам працівник). Після проведення розслідування заповнюється певна форма, в якій записуються вся інформація і результати розслідування. Роботодавець повинен нести облік всіх нещасних випадків, аварій, професійних захворювань тощо.

За порушення законодавства про охорону праці роботодавець буде притягнутий до сплати штрафу, в порядку встановленому законом. Важливо зауважити, сплата штрафу не знімає відповідальності і роботодавець все одно зобов'язаний впровадити норми охорони праці на підприємстві. Також таке відношення до організації праці може призвести до нещасних випадків, а це, в свою чергу, до кримінальна відповідальність. І, як вже було згадано раніше,

відповідальність за охорону праці несе роботодавець і відповідно при будь-якому порушенні правил охорони праці він є відповідальною особою.

Висновок, охорона праці – це правова система, яка створена для забезпечення безпеки працівників під час трудової діяльності. Включає в себе правила підготовки та організації праці. Зобов'язує роботодавця проводити ознайомлення з місцем праці, інструктаж, перевірку знань, атестацію робочого місця. Також охорона праці захищає здоров'я працівників: зобов'язує проведення періодичного медогляду та забезпечення працівників засобами індивідуального захисту (якщо такі потрібні). Будь-яке порушення охорони праці карається згідно закону.

3.2 Підвищення стійкості підприємства автомобілебудівної галузі у воєнний час

Під стійкістю роботи підприємства (в тому числі автомобілебудівного) розуміють здатність його в умовах надзвичайних ситуацій мирного і воєнного часу випускати продукцію в запланованому обсязі й номенклатурі, а при одержанні слабких і середніх руйнувань, порушенні зв'язків по кооперації і постачанням, відновлювати виробництво в мінімальний термін.

Здатність об'єкта автомобілебудівної галузі випускати продукцію залежить від захисту і нормального функціонування чотирьох основних елементів сучасного виробництва, якими є:

- виробничий персонал (робітники та службовці);
- будинки і споруди з технологічним устаткуванням;
- система постачання енергією, водою, паливом, устаткуванням і ремонтною базою;
- система виробничих і кооперативних зв'язків з іншими об'єктами.

Тому стійкість роботи об'єктів і галузі автомобілебудування в цілому в умовах надзвичайних ситуацій визначається наступними факторами:

- надійністю захисту робітників та службовців від усіх вражаючих факторів зброї масового ураження;
- здатністю інженерно-технічного комплексу (ІТК) об'єкта протистояти вражаючим факторам ядерного вибуху;
- надійністю системи постачання об'єкта всім необхідним для виробництва продукції (сировиною, паливом, що комплектують виробами, електроенергією, водою, газом тощо.);
- захищеності об'єкта від вторинних вражаючих факторів (пожеж, вибухів, затоплень, зараження місцевості отруйними і сильнодіючими отруйними речовинами);
- підготовленість об'єкта до проведення рятувальних та інших невідкладних робіт і робіт з відновленням порушеного виробництва.

Перераховані фактори визначають собою основні, загальні для всіх об'єктів промисловості, шляхи підвищення стійкості роботи в надзвичайних ситуаціях, а саме:

- забезпечення надійного захисту робітників та службовців від вражаючих факторів зброї масового ураження;
- забезпечення стійкості постачання всім необхідним для випуску запланованої на час надзвичайних ситуацій продукцією;
- підготовка до відновлення порушеного виробництва.

Захист робітників та службовців в умовах НС мирного і воєнного часу. Це найголовніша задача по підвищенню стійкості роботи об'єкта господарювання. Робітники й службовці – головна продуктивна сила і тому стійкість економіки визначається, насамперед, здатністю захистити і зберегти цю силу.

Військові конфлікти супроводжуються руйнуванням будинків, споруджень і знищенням основної продуктивної сили – працюючого населення. Тому серед усіх задач по підвищенню стійкості роботи об'єктів автомобілебудування основною є задача завчасного вживання заходів по забезпеченню захисту робітників та службовців і членів їхніх родин [73].

Захист робітників та службовців від зброї масової поразки в сучасних умовах здійснюється трьома основними способами:

- укриття людей у захисних спорудженнях (сховищах, протирадіаційних укриттях);
- проведення евакуації робітників, службовців і членів їхніх родин;
- використання засобів індивідуального захисту, а також проведенням заходів щодо протирадіаційного, протихімічного і проти бактеріологічного захисту з урахуванням конкретних обставин [72].

Варто також підкреслити, що найважливішою умовою успішного вирішення задачі захисту людей є навчання їх правилам дії по сигналах оповіщення цивільного захисту, застосуванню способів і засобів захисту, наданню самодопомоги і взаємодопомоги, діям у складі формувань ЦЗ.

Захист засобів виробництва. Такий захист полягає в підвищенні фізичної опірності будинків, споруджень і конструкцій об'єкта до впливу вражаючих факторів ядерного вибуху, захисту технологічного і верстатного устаткування, засобів зв'язку й інших засобів, що складають матеріальну основу виробничого процесу.

Методика оцінки стійкості будинків, технологічного устаткування об'єкта народного господарства до вражаючих факторів ядерного вибуху виконується по трьох основних вражаючих факторах:

- від впливу ударної хвилі ядерного вибуху;
- від світлового випромінювання на предмет виникнення пожеж;
- від радіації на предмет захисту виробничого персоналу від опромінення.

Забезпечення стійкого постачання підприємства. Для виробництва продукції необхідні: електроенергія, вода, паливо, сировина, матеріали й інші матеріально-технічні засоби. Забезпечення підприємств цими ресурсами багато в чому визначає можливість нормального їхнього функціонування в умовах воєнного часу. Це досягається проведенням таких заходів, що сприяють підвищенню не ураженості комунально-енергетичних мереж,

транспортних комунікацій і джерел постачання, надійному захисту необхідних запасів палива, сировини, напівфабрикатів, що комплектують, виробів тощо.

Підготовка до відновлення порушеного виробництва. Можливості вражаючою дії сучасних видів зброї такі, що забезпечити абсолютний захист від нього об'єктів і споруд практично неможливо. Вони можуть одержати той чи інший ступінь руйнування. У цих умовах задача зводиться до того, щоб у випадку слабких і середніх руйнувань на об'єкті відбудувати об'єкт і відновити випуск необхідної продукції в мінімальний термін.

Підвищення, стійкості роботи об'єкта автомобілебудівної галузі у воєнний час і в умовах надзвичайних ситуацій досягається завчасним проведенням комплексу інженерно-технологічних, технологічних і організаційних заходів, спрямованих на максимальне зниження впливу вражаючих факторів зброї масового ураження і створення умов для швидкої ліквідації наслідків. Підготовка до відновлення порушеного виробництва здійснюється завчасно і передбачає планування відбудовних робіт по декількох варіантах: підготовку ремонтних бригад, створення необхідного запасу матеріалів і устаткування, надійний його захист [72].

Висновок. Зросло значення економіки в забезпеченні обороноздатності країни. Реальна загроза руйнування її з використанням сучасних засобів нападу висувають у число основних задач підвищення стабільної роботи об'єктів автомобілебудівної галузі у воєнний час. Виконання цієї задачі покладається на сили цивільної захисту. Підвищення стійкості роботи досягається проведенням ряду організаційних та інженерно-технічних заходів, що враховують вимоги ЦЗ та БЖД. Результати оцінки стійкості роботи об'єкта і спрямованих на забезпечення безперебійності виробничого процесу в надзвичайній ситуації допомагають досягти максимального зниження можливих втрат і руйнувань.

ВИСНОВОК

Розвиток високопродуктивних комп'ютерів, здатних виконувати навчання та робити висновки для моделей машинного навчання, призводить до значного прогресу у нових підходах до відомих проблем. Однак для промислового застосування часто потрібні рішення, для машинного навчання, які можуть бути розгорнуті на обчислювально недорогих машинах, які мають низьку вартість і розмір. Розгортання моделей машинного навчання на низькопродуктивній апаратній платформі передбачає використання недорогих моделей з точки зору обчислювальної потужності та ресурсів пам'яті, що може бути досягнуто шляхом ретельного проектування архітектури моделі нейронної мережі. Паралельно з розвитком апаратного забезпечення, у розробці нових процесорних блоків, призначених для машинного навчання та, точніше, додатків глибокого навчання.

В першому розділі кваліфікаційної роботи:

- проаналізовано науково-технічні публікації які стосуються теми навчання та керування безпілотним авто за допомогою нейронної мережі;
- описано методи та принципи глибокого навчання;
- проаналізована набір готових даних для навчання нейронної мережі автономному водінню;
- описанні методи оптимізації моделі глибокого навчання.

В другому розділі кваліфікаційної роботи:

- проаналізовано існуючі модель штучних нейронних мереж для автономного водіння та запропоновано власну, J-Net.
- проведено тест в середовищі Unity існуючих моделей та J-Net.
- проведено аналіз результатів тесту.

Результат тестування: J-Net являється найменш ресурсозатратним рішенням серед інших моделей (які були проаналізовані в роботі), проте якісні результати моделі J-Net практично не відрізняються, а деколи навіть перевищують інші, що робить J-Net хорошим, практичним рішенням для

задачі навчання та керування безпілотним авто за допомогою нейронної мережі.

В третьому розділі кваліфікаційної «Охорона праці та безпека в надзвичайних ситуаціях» проаналізовано охорону праці на виробництві автомобільного транспорту, описано правила та норм праці які мають виконувати керівники, роботодавець та працівники підприємства. Також описано підвищення стійкості підприємства автомобілебудівної галузі у воєнний час.

Глибока нейронна мережа J-Net представлена в цій роботі, є одним із можливих рішень для наскрізного навчання автономного водіння. Метою нашої роботи було досягнення успішного автономного водіння за допомогою легкої глибокої нейронної мережі. Було запропоновано J-Net, глибоку нейронну мережу, здатну успішно виконувати завдання автономного водіння на репрезентативній доріжці (тестовий трек стимулятора Unity), при цьому обчислювальна вартість використання цієї моделі є найменшою серед інших відомих рішень які були також досліджується в цій роботі. Основний внесок запропонованої роботи - це нове рішення, яке є обчислювально ефективним завдяки відносно легкій архітектурі.

Обмеженням J-Net є недостатнє тренування в більш складних сценаріях. На додаток до цього, наша модель навчається з використанням необроблених зображень камери та вимірювання кута повороту для кожного кадру, тоді як швидкість автомобіля приймається за постійну (зумовлено віртуальним простором). Це інше сприйняття швидкості під час автономного руху, оскільки мається на увазі віртуальна швидкість яка надсилається моделі зразу в точних цифрах. Тим не менш J-Net можна навчити передбачати швидкість автомобіля. Подібним підходом як і прогнозуванням кута повороту керма, тобто, до одночасного прогнозування кута повороту добавиться швидкість на основі зображення вхідної камери в режимі реального часу.

Майбутня робота включатиме розгортання представленої мережі на вбудованій автомобільній платформі з обмеженими апаратними ресурсами, низькою потужністю процесора та малим об'ємом пам'яті. Можливі кінцеві варіанти використання представленої наскрізної навчальної мережі - це безпілотне таксі, транспортні засоби доставки тощо. Використання легкого рішення DNN, яке представлено в цьому проекті, дозволяє розгортати його на вбудованих автомобільних платформах з малопотужним обладнанням, низькою вартістю та розмірами, що важливо для практичного промислового застосування.

ПЕРЕЛІК ДЖЕРЕЛ

1. Боярський М., Дель Теста Д., Двораковський Д., Фірнер Б., Флепп Б., Гоял П., Жакель Л., Монфорт М., Мюллер У., Чжан Дж. Та ін. Навчання з кінця до кінця для автомобілів, що керують собою. arXiv. 2016. 1604.07316
2. Боярський М., Єрес П., Хороманська А., Хороманський К., Фірнер Б., Жакель Л., Мюллер У. Пояснюючи, як глибока нейронна мережа, навчена наскрізному навчання, керує автомобілем. arXiv. 2017. 1704.07911
3. Мехта А., Адітя С., Анбумані С. Навчання наскрізного автономного водіння за допомогою допоміжного нагляду під керівництвом. arXiv. 2018. 1808.10393
4. Чен Ю., Ван Дж., Лі Дж., Лу Ч., Ло З., Сюе Х., Ван С. Набір даних водіння LiDAR-Video: Ефективне навчання політиці керування автомобілем; Матеріали конференції IEEE/CVF 2018 року щодо комп'ютерного зору та розпізнавання шаблонів; Солт -Лейк -Сіті, Юта, США. 18–23 червня 2018 р .; С. 5870–5878. [[Google Scholar](#)]
5. Рамезані Дуракі А., Лі Д.-Ж. Наскрізний глибокий підсилюючий навчальний інтелектуальний агент, здатний до автономних досліджень у невідомих середовищах. Сенсори. 2018; 18: 3575. doi: 10.3390/s18103575. [[Безкоштовна стаття PMC](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]
6. Крижевський А., Суцкевер І., Хінтон Г. Є. Класифікація іміджевої мережі з глибокими згортковими нейронними мережами. В: Перейра Ф., Burges CJC, Bottou L., Weinberger KQ, ред. Досягнення нейронних систем обробки інформації. Фонд систем нейронної обробки інформації, Inc .; Ванкувер, Британська Колумбія, Канада: 2012. С. 1097–1105. [[Google Scholar](#)]
7. Udacity, Inc. Самостійний симулятор автомобіля. [(доступ 5 листопада 2018 р.)]; Доступно в Інтернеті: <https://github.com/udacity/self-driving-car-sim>.
8. Гудфеллоу І., Бенджіо Ю., Курвіль А. Глибоке навчання. Преса Массачусетського технологічного інституту; Кембридж, Массачусетс, США:

2017. [(доступ 5 листопада 2018 р.)]. Доступно Інтернеті:<https://www.deeplearningbook.org>. [[Google Scholar](#)]

9. Нейронні мережі та глибоке навчання Aggarwal СС. Міжнародне видавництво Springer; Чам, Швейцарія: 2018. [[Google Scholar](#)]

10. Чолет Ф. Глибоке навчання за допомогою Python. Публікації комплектування; Острів притулку, Нью -Йорк, США: 2018. [[Google Scholar](#)]

11. Саттон Р. С., Барто А. Г. Підсилення навчання. 2 -е вид. Преса Массачусетського технологічного інституту; Кембридж, Массачусетс, США: 2018. стор. 552. [[Google Scholar](#)]

12. LeCun Y., Boser B., Denker JS, Henderson D., Howard RE, Hubbard W., Jackel LD Backpropagation, застосовані до розпізнавання рукописного індексу. Нейронний комп'ютер. 1989; 1: 541–551. doi: 10.1162/neco.1989.1.4.541. [[CrossRef](#)] [[Google Scholar](#)]

13. ЛеКун Ю., Боттоу Л., Бенджо Ю., Хаффнер П. Градієнтне навчання, що застосовується до розпізнавання документів. [(доступ 1 травня 2019 р.)]; Доступно в Інтернеті:<http://yann.lecun.org/exdb/publis/pdf/lecun-01a.pdf>.

14. Simard D., Steinkraus PY, Platt JC Найкращі практики для згорткових нейронних мереж, що застосовуються до візуального аналізу документів; Матеріали сьомої Міжнародної конференції з аналізу та визнання документів; Единбург, Великобританія. 6 серпня 2003 р .; С. 958–963. [[Google Scholar](#)]

15. Шін Х., Рот Х., Гао М., Лу Л., Сюй З., Ногус І., Яо Дж., Моллура Д., Саммерс Р. Глибокі згорткові нейронні мережі для автоматизованого виявлення: архітектури CNN, Характеристики набору даних та трансферне навчання. IEEE Trans. Мед. Зображення. 2016; 35: 1285–1298. doi: 10.1109/TMI.2016.2528162. [[Безкоштовна стаття PMC](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]

16. Pathak D., Krähenbühl P., Donahue J., Darrell T., Efros AA Context Encoders: Feature Learning by Inpainting; Матеріали конференції IEEE 2016

року з комп'ютерного зору та розпізнавання моделей (CVPR); Лас -Вегас, штат Невада, США. 27–30 червня 2016 р .; С. 2536–2544. [[Google Scholar](#)]

17. Karpathy A., Toderici G., Shetty S., Leung T., Sukthankar R., Li F.-F. Широкомасштабна класифікація відео з згортковими нейронними мережами; Матеріали конференції IEEE 2014 року з комп'ютерного зору та розпізнавання моделей; Колумб, Огайо, США. 23–28 червня 2014 р .; С. 1725–1732. [[Google Scholar](#)]

18. Чі Дж., Кім Х.-К. Прогнозування концентрації льоду в Арктичному морі за допомогою глибокої нейронної мережі, керованої даними. Дистанційне зондування.2017; 9: 1305. [[Google Scholar](#)]

19. Руссаковський О., Денг Дж., Су Х., Краузе Дж., Сатіш С., Ма С., Берг А. С. Imagenet Велика проблема візуального розпізнавання. Int. J. Comput. Vis. 2015; 115: 211–252. doi: 10.1007/s11263-015-0816-у. [[CrossRef](#)] [[Google Scholar](#)]

20. Симонян К., Зіссерман А. Дуже глибокі згорткові мережі для широкомасштабного розпізнавання зображень. arXiv. 2004. 1409.1556

21. Сегеді С., Лю В., Цзя Ю., Серманет П., Рід С., Ангуелов Д., Ерхан Д., Ванхуке В., Рабінович А. Заглиблюючись із звивинами; Матеріали конференції IEEE з комп'ютерного зору та розпізнавання шаблонів; Бостон, Массачусетс, США. 7–12 червня 2015 р .; С. 1–9. [[Google Scholar](#)]

22. He K., Zhang X., Ren S., Sun J. Глибоке залишкове навчання для розпізнавання зображень; Матеріали конференції IEEE з комп'ютерного зору та розпізнавання шаблонів; Лас -Вегас, штат Невада, США. 27–30 червня 2016 р .; С. 770–778. [[Google Scholar](#)]

23. Вісін Ф., Кастнер К., Чо К., Маттеуччі М., Курвіль А., Бенджо Ю. Ренет: Альтернатива згортковим мережам на основі рекурентних нейронних мереж. arXiv. 2015. 1505.00393

24. Зоф Б., Васудеван В., Шленс Дж., Ле QV Навчання переданих архітектур для масштабованого розпізнавання зображень; Матеріали конференції IEEE/CVF 2018 року щодо комп'ютерного зору та розпізнавання

шаблонів; Солт -Лейк -Сіті, Юта, США. 18–23 червня 2018 р .; С. 8697–8710.

[\[Google Scholar\]](#)

25. Acuna D., Ling H., Kar A., Fidler S. Efficient Interactive Annotation of Segmentation Datas with Polygon-RNN ++; Матеріали конференції IEEE/CVF 2018 року щодо комп'ютерного зору та розпізнавання шаблонів; Солт -Лейк -Сіті, Юта, США. 18–23 червня 2018 р .; С. 859–868. [\[Google Scholar\]](#)

26. Ван Т.Ц., Лю М.Ю., Чжу Дж., Лю Г., Тао А., Каутц Дж., Катанзаро Б. Синтез відео-відео; Матеріали 32-ї Міжнародної конференції з систем нейронної обробки інформації; Монреаль, КК, Канада. 3–8 грудня 2018 р .; С. 1152–1164. [\[Google Scholar\]](#)

27. Сільвер Д., Шріттвізер Дж., Сімонян К., Антоноглу Дж., Хуан А., Гуез А., Губерт Т., Бейкер Л., Лай М., Болтон А. Освоєння гри ходити без відома людини. Природа. 2017; 550: 354. doi: 10.1038/природа24270. [\[PubMed\]](#) [\[CrossRef\]](#) [\[Google Scholar\]](#)

28. Сільвер Д., Хуберт Т., Шріттвізер Дж., Антоноглу І., Лай М., Гез А., Лілікрап Т. Освоєння шахів та сєгі за допомогою самостійної гри із загальним алгоритмом навчання. arXiv. 2017. 1712.01815

29. Amodei D., Ananthanarayanan S., Anubhai R., Bai J., Battenberg E., Case C., Chen J. Deep speech 2: Розпізнавання мовлення наскрізним англійською та мандаринською мовами; Матеріали 33-ї Міжнародної конференції з машинного навчання; Нью -Йорк, Нью -Йорк, США. 19–24 червня 2016 р .; С. 173–182. [\[Google Scholar\]](#)

30. Чен З., Чжан Т., Оуян С. Наскрізне виявлення літака за допомогою трансферного навчання у зображеннях дистанційного зондування. Remote Sens.2018; 10: 139. doi: 10.3390/rs10010139. [\[CrossRef\]](#) [\[Google Scholar\]](#)

31. Yao Y., Wang H., Li S., Liu Z., Gui G., Dan Y., Hu J. End-to-End згортова модель нейронної мережі для діагностики несправностей передач на основі звукових сигналів. Заявка Наук. 2018; 8: 1584. [\[Google Scholar\]](#)

32. Kanade T., Thorpe C., Whittaker W. Проект автономного наземного транспортного засобу в КМУ; Матеріали чотирнадцятої щорічної

конференції ACM з інформатики 1986 р. Цинциннаті, Огайо, США. 4–6 лютого 1986 р. [[Google Scholar](#)]

33. Уоллес Р. Перші результати роботи-слідування по дорозі; Матеріали 9-ї міжнародної спільної конференції з питань штучного інтелекту; Лос-Анджелес, Каліфорнія, США. 18–23 серпня 1985 р. [[Google Scholar](#)]

34. Дікманнс Е. Д., Запп А. Автономне керівництво високошвидкісним дорожнім транспортним засобом за допомогою комп'ютерного бачення. IFAC Proc. Вип. 1987; 20: 221–226. doi: 10.1016/S1474-6670 (17) 55320-3. [[CrossRef](#)] [[Google Scholar](#)]

35. Thrun S., Montemerlo M., Dahlkamp H., Stavens D., Aron A., Diebel J., Fong P., Gale J., Halpenny M., Hoffmann G. et al. Стенлі: Робот, який переміг у великому виклику DARPA. Дж. Польовий робот. 2006; 23: 661–692. doi: 10.1002/rob.20147. [[CrossRef](#)] [[Google Scholar](#)]

36. Монтемерло М., Трун С., Далкамп Х., Ставенс Д., Стробанд С. Перемога у грандіозному виклику DARPA за допомогою робота -штучного інтелекту; Матеріали 21-ї національної конференції з питань штучного інтелекту; Бостон, Массачусетс, США. 16–20 липня 2006 р .; С. 982–987. [[Google Scholar](#)]

37. Buehler M., Iagnemma K., Singh S. Springer Tracts in Advanced Robotics. Спрингер; Берлін/Гейдельберг, Німеччина: 2009. Міський виклик DARPA: Автономні транспортні засоби у міському русі. [[Google Scholar](#)]

38. Сюй Д., Джайн А., Ангуелов Д. PointFusion: Глибоке злиття сенсорів для оцінки 3D обмежувальної рамки; Матеріали конференції IEEE/CVF 2018 року щодо комп'ютерного зору та розпізнавання шаблонів; Солт -Лейк -Сіті, Юта, США. 18–23 червня 2018 р .; С. 244–253. [[Google Scholar](#)]

39. О С., Кан Х. Швидка фільтрація сітки з використанням мережевих кластерів із даних LIDAR та даних стереовізора. IEEE Sens. J. 2016; 16: 7258–7266. doi: 10.1109/JSEN.2016.2598600. [[CrossRef](#)] [[Google Scholar](#)]

40. Чавес-Гарсія Р. О., Айкард О. Злиття та класифікація кількох сенсорів для виявлення та відстеження рухомих об'єктів. IEEE Trans. Інтелл. Трансп. Сист. 2016; 17: 525–534. doi: 10.1109/TITS.2015.2479925. [[CrossRef](#)] [[Google Scholar](#)]

41. Чо Х., Се Ю., Вія Кумар БВК, Раджкумар Р. Р. Багатоядерна термоядерна система для виявлення та відстеження рухомих об'єктів у міських умовах водіння; Матеріали Міжнародної конференції IEEE 2014 року з робототехніки та автоматизації (ICRA); Гонконг, Китай. 31 травня - 7 червня 2014 р .; С. 1836–1843. [[Google Scholar](#)]

42. Раванкар А., Раванкар А. А., Кобаяші Ю., Хошино Ю., Пенг С.-К. Методи згладжування шляхів у навігації роботів: найсучасніші, сучасні та майбутні виклики. Сенсори. 2018; 18: 3170. doi: 10.3390/s18093170. [[Безкоштовна стаття PMC](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]

43. Вей К., Рен Б. Метод динамічного планування шляхів автономного уникнення перешкод роботизованим маніпулятором на основі вдосконаленого алгоритму RRT. Сенсори. 2018; 18: 571. doi: 10.3390/s18020571. [[Безкоштовна стаття PMC](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]

44. Cai W., Zhang M., Zheng YR Призначення завдань та планування маршруту для кількох автономних підводних транспортних засобів за допомогою 3D-кривих Дубіна. Сенсори. 2017; 17: 1607. doi: 10.3390/s17071607. [[Безкоштовна стаття PMC](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]

45. Паден Б., Чап М., Йонг С.З., Єршов Д., Фраццолі Е. Огляд методів планування руху та керування для міських транспортних засобів, що керують автомобілем. IEEE Trans. Інтелл. Veh. 2016; 1: 33–55. doi: 10.1109/TIV.2016.2578706. [[CrossRef](#)] [[Google Scholar](#)]

46. Гонсалес Д., Перес Ж., Міланес В., Нашашібі Ф. Огляд методів планування руху для автоматизованих транспортних засобів. IEEE Trans.

Интелл. Трансп. Сист. 2016; 17: 1135–1145. doi: 10.1109/TITS.2015.2498841.

[CrossRef] [Google Scholar]

47. Сун Ю., Джин Ю., Квак Дж., Лі С.-Г., Чо К. Розширений підхід обрізання зображення з камери для наскрізного керування на стійкій обчисленні на основі CNN. Стійкість. 2018; 10: 816. doi: 10.3390/su10030816.

[CrossRef] [Google Scholar]

48. Косіć J., Jovićić N., Drndarević V. Клонування поведінки водія за допомогою глибокого навчання; Матеріали 17-го Міжнародного симпозиуму INFOTEN-JAHORINA (INFOTEN); Східне Сараєво, Республіка Сербська. 21–23 березня 2018 р .; С. 1–5. [Google Scholar]

49. Наварро А., Йерденінг Дж., Халіл Р., Браун А., Ашер З. Розробка стратегії автономного управління транспортними засобами з використанням однієї камери та глибоких нейронних мереж. SAE; Warrendale, PA, США: 2018. Технічний документ SAE № 2018-01-0035. [Google Scholar]

50. Рідміллер М., Монтемерло М., Далкамп Х. Навчання керуванню справжнім автомобілем за 20 хвилин; Матеріали «Кордони 2007 року у конвергенції біологічної науки та інформаційних технологій»; Місто Чеджу, Корея. 11–13 жовтня 2007 р .; С. 645–650. [Google Scholar]

51. Ву К., Аболфазлі Есфакані М., Юань С., Ван Х. Навчіться керувати шляхом глибокого посилення навчання. Сенсори. 2018; 18: 3650. doi: 10.3390/s18113650. [Безкоштовна стаття PMC] [PubMed] [CrossRef] [Google Scholar]

52. Shalev-Shwartz S., Shammah S., Shashua A. Безпечне, багатоагентське, навчання підкріплення для автономного водіння. arXiv. 2016. 1610.03295

53. Фрідман Л., Дженік Б., Тервіллігер Дж. Глибокий трафік: Швидка їзда через щільний рух з глибоким навчанням підкріплення. arXiv. 2018. 1801.02805

54. Лі К.В., Юн Х.С., Сонг Дж.М., Парк К.Р. Класифікація емоцій водія під час агресивного та плавного водіння за допомогою мультимодальних

сенсорів камери на основі згорткової нейронної мережі. Сенсори. 2018; 18: 957. doi: 10.3390/s18040957. [[Безкоштовна стаття PMC](#)] [[PubMed](#)] [[CrossRef](#)] [[Google Scholar](#)]

55. Фрідман Л. Автономні транспортні системи, орієнтовані на людину: принципи ефективної спільної автономії. arXiv. 2018. 1810.01835

56. Теті М., Баренгольц Е., Мартін С., Хан В. Систематичне порівняння архітектур глибокого навчання в автономному транспортному засобі. arXiv. 2018. 1803.09386

57. Kocić J., Jovičić N., Drndarević V. Sensors and Sensor Fusion in Autonomous Vehicles; Матеріали 26-го телекомунікаційного форуму 2018 року (TELFOR); Белград, Сербія. 20–21 листопада 2018 р .; С. 420–425. [[Google Scholar](#)]

58. Іандола Ф., Кейтцер К. Основний доповідь: малі нейронні мережі прекрасні: забезпечують вбудовані системи з невеликими архітектурами глибоких нейронних мереж; Матеріали Міжнародної конференції 2017 року з кодекстру апаратного/програмного забезпечення та синтезу систем; Сеул, Корея. 15–20 жовтня 2017 р .; С. 1–10. [[Google Scholar](#)]

59. Глорот Х., Бенджіо Ю. Розуміння труднощів навчання глибоких нейронних мереж упередженого характеру; Матеріали 13-ї Міжнародної конференції зі штучного інтелекту та статистики; Сардинія, Італія. 13–15 травня 2010 р. [[Google Scholar](#)]

60. Орпоне П. Обчислювальна складність нейронних мереж: опитування. Північ. J. Comput. 1994; 1994: 94–110. [[Google Scholar](#)]

61. Рагу М., Пул Б., Кляйнберг Дж., Гангулі С., Сол Дікштейн Дж. Про виразну силу глибоких нейронних мереж; Матеріали 34-ї Міжнародної конференції з машинного навчання; Сідней, Австралія. 6–11 серпня 2017 р .; С. 2847–2854. [[Google Scholar](#)]

62. Се В., Чен Ю., Ян Т., Емер Е. С. Ефективна обробка глибоких нейронних мереж: Навчальний посібник та опитування. Proc. IEEE. 2017; 105: 2295–2329. doi: 10.1109/JPROC.2017.2761740. [[CrossRef](#)] [[Google Scholar](#)]

63. Косіс J. Модель Shallow. [(доступ 22 лютого 2019 р.)]; Доступно в Інтернеті:<https://www.youtube.com/watch?v=w4UUz-gI7yw>.
64. Гуллі А., Пал С. Глибоке навчання з Керасом. Packt Publishing; Бірмінгем, Великобританія: 2017. [[Google Scholar](#)]
65. Кінгма Д. П., Ба Дж. Адам: Метод стохастичної оптимізації; Матеріали 3 -ї Міжнародної конференції з навчальних представництв; Сан - Дієго, Каліфорнія, США. 7–9 травня 2015 р .; С. 1–15. [[Google Scholar](#)]
66. Косіс J. Модель AlexNet. [(доступ 5 листопада 2018 р.)]; Доступно в Інтернеті:https://www.youtube.com/watch?v=ICxRQfm5a_4.
67. Косіс J. Model PilotNet (NVIDIA Net) [(доступ 5 листопада 2018 р.)]; Доступно в Інтернеті:<https://www.youtube.com/watch?v=YPZRKh4xbm4>.
68. Косіс J. Модель J-Net. [(доступ 5 листопада 2018 р.)]; Доступно в Інтернеті:<https://www.youtube.com/watch?v=BB-YljzloWI>.
69. Закон України «Про охорону праці» редакція від 14.08.2021.
70. Типове положення «Про службу охорони праці» НПАОП 0.00-4.35-04.
71. Кодекс законів про працю України.
72. Указ президента України «Про концепцію захисту населення і територій у разі загрози та виникнення надзвичайних ситуацій» (Указ від 26.03.1999 № 284/99)
73. Кодекс цивільного захисту України

ДОДАТКИ

Тези конференції

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

ІХ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



8-9 грудня 2021 року

**ТЕРНОПІЛЬ
2021**

В.О. Колодій, В.Г. Онущакій АНАЛІЗ МЕТОДІВ ДОСЛІДЖЕННЯ ВІБРАЦІЙНОЇ СТІЙКОСТІ ЕЛЕКТРОУСТАНОВОК V.O. Kolodiy, V.G. Onutsky ANALYSIS OF METHODS FOR STUDYING THE VIBRATION RESISTANCE OF ELECTRICAL INSTALLATIONS	49
О.О. Ліщук, Д.А. Радчук, Т.Б. Зошук РОЗУМНІ МІСТА ТА ІНТЕРНЕТ РЕЧЕЙ O.O. Lishchuk, D.A. Radchuk, T.B. Zoshchuk SMART CITIES AND THE INTERNET OF THINGS	50
Д.І. Матіюк, В.В. Никитюк ОНЛАЙН-ІНСТРУМЕНТ GOOGLE SHEETS ДЛЯ СИСТЕМАТИЗОВАНИХ КОНСОЛІДОВАНИХ ДАНИХ ВАКЦИНАЦІЇ НЕМОВЛЯТ D. Matsyuk, V. Nykytyuk GOOGLE SHEETS ONLINE TOOL FOR SYSTEMATIZED CONSOLIDATED INFAN VACCINATION DATA	51
М. Мандзій, І. Поліщук, П. Контсогограда, І. Дедів ЗАДАЧА ОПТИМАЛЬНОГО ВИЯВЛЕННЯ СИГНАЛІВ В СУМІШІ ІЗ ЗАВАДАМИ В ОБЛАСТІ РАДІОТЕХНІКИ M. Mandziy, I. Polishchuk, P. Kotsyograda, I. Dediv THE PROBLEM OF OPTIMAL DETECTION OF SIGNALS IN MIXTURE WITH INTERFERENCES IN THE FIELD OF RADIO ENGINEERING	52
Л. Матійчук, І. Павлов, В. Сташук ТЕОРЕТИЧНЕ ОБГРУНТУВАННЯ МЕТОДУ ВИЯВЛЕННЯ КОМП'ЮТЕРНИХ АТАК L. Matiyechuk, I. Pavlov, V. Stashuk THEORETICAL JUSTIFICATION OF THE METHOD OF DETECTION OF COMPUTER ATTACKS	53
Л. Матійчук, І. Павлов, В. Сташук ОЦІНКА ІСНУЮЧИХ СИСТЕМ ВИЯВЛЕННЯ АТАК L. Matiyechuk, I. Pavlov, V. Stashuk EVALUATION OF EXISTING ATTACK DETECTION SYSTEMS	55
А.В. Мельничук МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ В РАМКАХ ПРЕДМЕТНО- ОРІЄНТОВАНОГО ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ A.V. Melnychuk INFORMATION PROTECTION METHODS WITHIN DOMAIN-DRIVEN DESIGN OF THE INFORMATION SYSTEM	57
М.В. Михайлів ПОПЕРЕДНЯ ОБРОБКА ВІДЕОЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ M.V. Mykhayliv PRE-PROCESSING OF VIDEO IMAGES USING NEURAL NETWORKS	58
О. Данильців, А. Хом'як, Т. Назаревич ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ДОСЛІДЖЕННЯ СТАНУ РОСЛИН В РОЗУМНИХ ТЕПЛИЦЯХ O. Danulytsiv, A. Khomiak, T. Nazarevych THE USE OF NEURAL NETWORKS FOR STUDY THE CONDITION OF PLANTS IN SMART GREENHOUSES	59

УДК 004.6

О.О. Ліщук, Д.А. Радчук ст.гр. СНм-61, Т.Б. Зошук ст.гр.СТм-61

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

РОЗУМНІ МІСТА ТА ІНТЕРНЕТ РЕЧЕЙ

UDC 004.6

O.O. Lishchuk, D.A. Radchuk, T.B. Zoshchuk

SMART CITIES AND THE INTERNET OF THINGS

Концепція розумних міст, що формується, стає вагомим прикладом того, як інформаційні технології можуть покращити якість життя при оптимізації міської діяльності. Оскільки понад половина світового населення проживає в містах та стрімкий ріст населення в країнах із економікою, що розвивається, існує вплив на перепланування існуючих міст та проектування нових міст з нуля, щоб стати зеленими та ефективними, забезпечивши транспортні системи, енергетичні мережі та державні служби, які забезпечать життєдіяльність мешканців міст.

Існує ряд сучасних технологій, еволюція та розгортання яких сприяє зростанню розумних міст. Розвиток розумних міст зосереджено на конкретних потребах інфраструктури, наприклад, зменшенні втрат води через старіша інфраструктури труб, підвищення ефективності перевезень тощо. Різні регіони мають різні потреби: Однак основні технологічні тенденції не відрізняються, і тому виникає необхідність застосування інформаційних технологій для задоволення потреб міста. Потрібно визначити свою роль в системі рішень розумного міста та працювати над розвитком партнерських відносин, які дозволять колективно пропонувати рішення для міст. Міста можуть запровадити проміжне програмне забезпечення та хмарні системи для збору та використання даних, які відбираються з різноманітних датчиків встановлених на території міста. Зазначимо, що сьогодні мало хто з міст збирає та всебічно аналізує дані міст.

Конфіденційність – це ще одна важлива проблема. Багато громадян турбуються про конфіденційність розумних лічильників. Менше споживання енергії може означати, що мешканець не перебуває вдома. Електронні медичні записи є величезним ризиком конфіденційності, як показав досвід *healthcare.gov*. Набагато більше зусиль щодо розбудови довіри на основі захисту конфіденційності та безпеки даних має відбутися до того, як розумні міста отримують одобрення громадян.

Позитивним є те, що концепція розумного міста, схоже, набуває визнання, принаймні у державних та технологічних компаніях. Прогнозують, що до 2020 року, але деякі дослідники стверджують, що це новаторство програми інтелектуальних ІКТ не можуть автоматично створити розумне місто.

Література.

1. Дуда О. М., Кунавець Н. Е., Мацюк О. В., Пасічник В. В. Системні комплекси інформаційних технологій у проєктах «Розумне місто» // Системний аналіз та інформаційні технології: матеріали 18-ї Міжнародної науково-технічної конференції SAIT 2016 / Київ: ННК «ПСА», 2016. – С. 215–216.
2. Дуда О. М., Кунавець Н. Е., Мацюк О. В., Пасічник В. В. Концепт «розумне місто» та інформаційні технології BigData // Матеріали V науково-технічної конференції „Інформаційні моделі, системи та технології“, Тернопіль, 2018. – С. 30.

Н.С. Таванець, В.В. Нукитюк СПОСІБ ГОЛОСОВОЇ ІДЕНТИФІКАЦІЇ КОРИСТУВАЧА N. Tavanets, V. Nykytyuk THE METHOD OF VOICE IDENTIFICATION OF THE USER	84
М. Тимцін, О. Яскілка АНАЛІЗ ЗАСТОСУНКІВ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ M. Tymkiv, O. Yaskilka ANALYSIS OF APPLICATIONS FOR THE STUDY OF FOREIGN LANGUAGES	85
Д.Ю. Дзюба, Л.П. Дмитроца ПРИСТРОЇ З ФІЗИЧНОЮ НЕКЛОНОВАНОЮ ФУНКЦІЄЮ (PUF) D. Dziuba, L. Dmytrotsa DEVICES WITH PHYSICAL NON-CLONED FUNCTION (PUF)	86
Ж.Ж. Захест, В.Б. Савкін РОЗРОБКА ТА ДОСЛІДЖЕННЯ СИСТЕМИ ВІРТУАЛЬНОГО ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ПРАЦІВНИКІВ ВИРОБНИЦТВ НА БАЗІ VR,AR ТА AI J. Zakhem, V. Savkin DEVELOPMENT AND RESEARCH OF A VIRTUAL SECURITY SYSTEM FOR PRODUCTION WORKERS BASED WITH THE HELP OF VR,AR AND AI	87
Т. Скуржанський, О.Б. Назаревич ОДНОШАРОВИЙ ПЕРЦЕПТРОН ЯК ІНСТРУМЕНТ ДЛЯ АНАЛІЗУ ГАЗОСПОЖИВАННЯ T. Skurzhanskyi, O. Nazarevych SINGLE LAYER PERCEPTRON AS A TOOL FOR GAS CONSUMPTION ANALYSIS	88
Д. Корж, Д. Радчук, М. Тимцін, А. Колесник, Т. Зошук РІЗНИЦЯ МІЖ «ТРАДИЦІЙНИМИ» ТА «РОЗУМНИМИ» МІСТАМИ D. Korzh, D. Radchuk, M. Tymkiv, A. Kolesnyk, T. Zoshchuk THE DIFFERENCE BETWEEN "TRADITIONAL" AND "SMART" CITIES	90
Д. Корж, Д. Радчук, О. Лішук, А. Колесник, Т. Зошук РОЗУМНА СИСТЕМА ЕЛЕКТРОННОГО ЗДОРОВ'Я ДЛЯ ВІДСТЕЖЕННЯ ТА МОНИТОРИНГУ ПАЦІЄНТІВ, ПЕРСОНАЛУ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ D. Korzh, D. Radchuk, O. Lishchuk, A. Kolesnyk, Zoshchuk T. SMART ELECTRONIC HEALTH SYSTEM FOR TRACKING AND MONITORING OF PATIENTS, PERSONNEL IN REAL TIME	92
М. Тимцін, О. Яскілка АНАЛІЗ ЗАСТОСУНКІВ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ M. Tymkiv, O. Yaskilka ANALYSIS OF APPLICATIONS FOR THE STUDY OF FOREIGN LANGUAGES	91
О.І. Тимчак, І.Ю. Дедів АЛГОРИТМ ВІДЛЕННЯ ТА РОЗПІЗНАВАННЯ ОБЛИЧЧЯ O.I. Tymchak, DediV. ALGORITHM OF SELECTION AND FACE RECOGNITION	93

УДК 004.6

Д. Корж – ст. гр. СНмз-61, Д. Радчук, О. Ліщук – ст. гр. СНм-61, А. Колесник,
Т. Зошчук – ст. гр. СТм-61,
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

РОЗУМНА СИСТЕМА ЕЛЕКТРОННОГО ЗДОРОВ'Я ДЛЯ ВІДСТЕЖЕННЯ ТА МОНІТОРИНГУ ПАЦІЄНТІВ, ПЕРСОНАЛУ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

UDC 004.6

D. Korzh, D. Radchuk, O. Lishchuk, A. Kolesnyk, Zoshchuk T.

SMART ELECTRONIC HEALTH SYSTEM FOR TRACKING AND MONITORING OF PATIENTS, PERSONNEL IN REAL TIME

Охорона здоров'я в Україні відстає від розвинених країн світу через недостатню кількість медичних працівників та відсутність застосування інформаційних технологій відстеження та моніторингу. Ця спричинило такі проблеми, як неправильна ідентифікація пацієнтів, довгий час очікування пацієнтів та неможливість ефективно використовувати медичне обладнання.

Україна повинна адаптуватися до вимог сучасної охорони здоров'я. Аналіз публікацій показав, що системи інформаційних технологій почали впроваджуватися в деякі лікарні, але навіть у цих лікарнях ці технології використовуються недостатньо.

Метою цієї публікації є надання відповідного вибору технології відстеження та моніторингу в реальному часі в охороні здоров'я у формі інтегрованої системи RFID/ZigBee. Така система має цілісну структуру для закладів охорони здоров'я, якої слід дотримуватися для індивідуальних рішень для підвищення ефективності та продуктивності персоналу, а також для кращого догляду за пацієнтами та мінімізації довгострокових витрат.

Структура включає в себе контекстуальні елементи як із трикутника стратегії інформаційної системи (ISST), так і з систем факторів відповідності людини, організації та технології (HOT-fit), таким чином, що нова структура враховує технологічні, організаційні, людські та бізнесові фактори.

Були проаналізовані різні випадки, щоб покращити робочий процес лікарень, використовуючи запропоновану технологію, включаючи такі процеси, як переміщення персоналу та медичних засобів. Це призвело до необхідності візуалізація та управління знаннями для підтримки аналізу даних у реальному часі для прийняття рішень бізнес-аналітики.

Кінцевою метою цього аналізу є надання інтерактивних платформ для медичного персоналу для підвищення ефективності та продуктивності.

Результатом цих удосконалень буде забезпечення кращого догляду за пацієнтами, скорочення часу очікування пацієнтів, зниження витрат на медичне обслуговування та надання більше часу персоналу для надання покращеної допомоги, орієнтованої на пацієнта, у секторі охорони здоров'я.

Література.

1. Hameed, R.T., Mohamad, O.A. & Tâpuș, N. (2016). Health Monitoring System Based on Wearable Sensors and Cloud Platform. 20th International Conference on System Theory, Control and Computing (ICSTCC). p.pp. 543–548.