
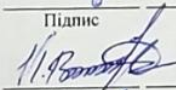
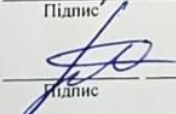



Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод розпізнавання дорожніх знаків за допомогою моделей
глибокого навчання

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент групи КН-21-2  Дмитрій ЛАЗАР
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ
Керівник: асистент кафедри КН  Леонід ВОЗНЮК
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ
Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
зав. кафедри КН, д.т.н., професор  Олександр БАРМАК
Підпис Ім'я, ПРІЗВИЩЕ
20 06 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
 Факультет інформаційних технологій
 Кафедра комп'ютерних наук
 Освітній ступінь бакалавр
 Галузь знань 12 – Інформаційні технології
 Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор Олександр БАРМАК

« 10 » 02 2025 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання»

2. Завдання видано студенту Дмитрій ЛАЗАР
 (ім'я, прізвище)

3. Керівник роботи Асистент кафедри КН: Леонід ВОЗНЮК
 (посада, ім'я, прізвище)

4. Затверджено наказом університету від « 07 » 02 2025 р. № 23

5. Дата видачі завдання студенту: « 10 » 02 2025 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метою кваліфікаційної роботи бакалавра є підвищення точності автоматичного розпізнавання дорожніх знаків на основі моделей глибокого навчання. Поставлену мету досягають шляхом виконання таких завдань: проведення аналізу нейромережових методів класифікації зображень; розробити метод розпізнавання дорожніх знаків з використанням згорткової нейронної мережі ResNet50; спроектувати архітектуру програмного забезпечення з підтримкою обробки відеопотоку в реальному часі; реалізувати інтелектуальний метод розпізнавання з графічним інтерфейсом; провести експериментальне тестування методу на основі відкритого датасету та відеозаписів з камери мобільного пристрою.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2025	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	лютий 2025	виконано
3	Проектування та розробка загальної архітектури програмного забезпечення, інтерфейсу користувача, вибір засобів реалізації програмного забезпечення	березень 2025	виконано
4	Створення та тестування програмного забезпечення	квітень 2025	виконано
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення згідно вимог	травень 2025	виконано
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2025	виконано
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2025	виконано
8	Підготовка до захисту та захист кваліфікаційної роботи бакалавра	червень 2025	виконано

Виконавець: студент групи КН-21-2
Група виконавця

Підпис

Дмитрій ЛАЗАР
Ім'я, ПРІЗВИЩЕ

Керівник: асистент кафедри КН
Науковий ступінь, посада

Підпис

Леонід ВОЗНЮК
Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-21-2 Дмитрій ЛАЗАР

Керівник кваліфікаційної роботи бакалавра: асистент кафедри КН: Леонід ВОЗНЮК


Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
47	17	8	43	2

Метою кваліфікаційної роботи бакалавра є підвищення точності розпізнавання дорожніх знаків шляхом проектування методу на основі згорткової нейронної мережі ResNet50. Для розробки програмного застосунку буде використано мову програмування Python і бібліотеки PyTorch, OpenCV, Tkinter. Створений застосунок має автоматично виявляти дорожні знаки на зображеннях із камери, класифікувати їх та виводити назви розпізнаних знаків із рівнем достовірності. Реалізація цього рішення має потенціал застосування в навчальних, дослідницьких і прикладних транспортних системах.

Ключові слова: розпізнавання дорожніх знаків, глибоке навчання, ResNet50, згорткова нейронна мережа.

Виконавець: студент групи КН-21-2
Група виконавця


Підпис

Дмитрій ЛАЗАР
Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень.....	7
Вступ.....	8
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій.....	10
1.1 Аналіз предметної області.....	10
1.2 Огляд теоретичних підходів до розв’язку подібних задач	11
1.3 Аналіз існуючих програмних засобів та наукових рішень	12
1.5 Мета, задачі та вимоги до реалізації інформаційної системи.....	17
1.6 Висновки до розділу 1.....	18
Розділ 2 Розроблення методу.....	19
2.1 Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання.....	19
2.2 Функціональна структура інформаційної системи.....	22
2.3 Архітектура нейронної мережі	23
2.4 Проектування архітектури інформаційної системи та взаємозв’язок компонентів	26
2.5 Підготовка датасету для нейромережевого розпізнавання дорожніх знаків.....	29
2.6 Особливості використання спеціалізованих програмних компонентів	30
2.7 Оцінювання точності розпізнавання дорожніх знаків	32
2.8 Висновки до розділу 2.....	33
У результаті в цьому розділі сформовано технічну та методологічну базу, необхідну для подальшої реалізації та тестування системи розпізнавання дорожніх знаків.Розділ 3 Експериментальне дослідження методу та програмна реалізація інформаційної системи.....	34
3.1 Визначення шляхів дослідження та засобів створення програмного забезпечення.....	35
3.2 Засоби розробки програмного застосунку.....	36
3.3 Структура та функціональне призначення програмних складових системи.....	38
3.4 Особливості реалізації програмних складових системи	40
3.5 Тестування програмного застосунку та вимоги до розгортання	44
3.6 Результати досліджень.....	48
3.7 Висновки до розділу 3.....	52
Загальні висновки.....	53
Перелік посилань	55
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
GTSRB	german traffic sign recognition benchmark
ACNN	convolutional neural network
КРБ	Кваліфікаційна робота бакалавра
ADAS	Advanced driver-assistance system
PIL	Python Imaging Library
CNN	Cable News Network
VS	Visual Studio
MFC	Microsoft Foundation Class
CPU	Central processing unit
CSV	Comma-separated values
GUI	Graphical user interface
GPU	Graphics processing unit

Вступ

Актуальність. У сучасному світі стрімкий розвиток транспортної індустрії диктує потребу у впровадженні передових технологій, які сприяють підвищенню рівня безпеки, комфорту та ефективності дорожнього руху. Важливим компонентом інфраструктури є дорожні знаки, що регулюють рух, інформують водіїв і знижують ризик виникнення аварій. Проте збільшення кількості транспортних засобів, складні умови дорожнього руху, а також погіршення стану дорожніх знаків через вплив навколишнього середовища створюють суттєві виклики для їх своєчасного розпізнавання водіями.

У зв'язку з цим все більш актуальною стає автоматизація процесу розпізнавання дорожніх знаків. Сучасні методи глибокого навчання, зокрема згорткові нейронні мережі, демонструють чудові результати в аналізі зображень, розпізнаванні об'єктів та їх класифікації. Завдяки здатності адаптуватися до різних умов освітлення, ракурсів і фонових змін, ці моделі ефективно справляються із завданням і можуть бути інтегровані в інтелектуальні транспортні рішення, такі як системи автопілотування, чи відеоспостереження.

Крім того, розробка таких технологій має велике значення в рамках цифрової трансформації транспортної інфраструктури та розвитку автономного транспорту. Таким чином, застосування методів глибокого навчання для розпізнавання дорожніх знаків є не лише науково перспективним, але й важливим соціальним внеском у створення безпечнішого та технологічно досконалого дорожнього середовища.

Об'єкт дослідження – процес розпізнавання дорожніх знаків на зображеннях отриманих при різних умовах освітлення та середовища.

Предмет дослідження – методи комп'ютерного зору та глибинного навчання для розпізнавання об'єктів на зображеннях.

Мета кваліфікаційної роботи бакалавра – є підвищення точності розпізнавання дорожніх знаків шляхом проєктування методу на основі згорткової нейронної мережі ResNet50

Завдання кваліфікаційної роботи бакалавра:

Для досягнення мети дослідження, яка стосується методів розпізнавання дорожніх знаків за допомогою моделей глибокого навчання, визначено такі завдання:

- Провести огляд існуючих методів розпізнавання дорожніх знаків через аналіз їхніх зображень, визначивши їх сильні та слабкі сторони.
- Дослідити переваги й недоліки застосування інтелектуальних методів аналізу даних для ідентифікації дорожніх знаків, враховуючи різноманітні умови та виклики.
- Розробити метод автоматичного розпізнавання дорожніх знаків, використовуючи сучасні підходи глибокого навчання.
- Провести тестування методу на різноманітних наборах даних.

Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій

1.1 Аналіз предметної області

У цьому розділі розглянуто ключові аспекти предметної області, що безпосередньо впливають на реалізацію кваліфікаційної роботи бакалавра. Зосереджено увагу на актуальних методах розпізнавання об'єктів на зображеннях, які мають вирішальне значення для побудови ефективних інформаційних систем. Однією з найважливіших складових безпечної дорожньої інфраструктури є здатність швидко й надійно ідентифікувати дорожні знаки, оскільки вони передають критично важливу інформацію водієві та системам допомоги водінню[1].

З поширенням автоматизованих транспортних засобів та розумних систем керування автомобілями зростає потреба в точному машинному розпізнаванні знаків у режимі реального часу[2]. Хоча раніше для цього застосовували традиційні алгоритми обробки зображень – сегментацію, фільтрацію, виявлення країв – вони демонстрували обмежену надійність у складних умовах, таких як тінь, зміни освітлення чи часткове перекриття знаків[3].

Сучасні підходи на основі глибокого навчання, зокрема згорткових нейронних мереж, дають змогу автоматично виділяти релевантні ознаки із зображень без ручного втручання[4]. Завдяки використанню спеціалізованих наборів даних, таких як GTSRB, моделі можуть ефективно навчатися розпізнавати широкий спектр знаків у різноманітних умовах[5].

Хоча успішність таких систем залежить від багатьох факторів – як-от обсяг даних, вибір архітектури, параметри тренування – вони демонструють високу точність та гнучкість у роботі. Аналіз предметної області підтверджує доцільність використання нейромережових методів для автоматизації задач розпізнавання дорожніх знаків. Попри певні труднощі реалізації, переваги таких рішень очевидні: від підвищення безпеки дорожнього руху до вдосконалення транспортної аналітики та інфраструктури.

Отже, проведений аналіз підтвердив, що використання глибокого навчання є найбільш перспективним напрямом для задач розпізнавання дорожніх знаків. Застосування згорткових нейронних мереж дозволяє значно підвищити точність класифікації та забезпечити стійку роботу системи в умовах реального середовища. Це створює надійну основу для подальшої реалізації програмної системи та досягнення практичних результатів.

1.2 Огляд теоретичних підходів до розв'язку подібних задач

Глибоке навчання стало основою розпізнавання дорожніх знаків, зокрема завдяки згортковим нейронним мережам (CNN), які дозволяють точно виявляти та класифікувати об'єкти без ручного проектування ознак. Найчастіше для навчання використовують датасет GTSRB, на базі якого створено архітектури LeNet-5, ResNet, MobileNet і YOLO[6]. ResNet забезпечує стабільність при великій глибині, MobileNet підходить для слабких пристроїв, а YOLO дозволяє виконувати виявлення й класифікацію в реальному часі. Хоч деякі моделі, як-от Faster R-CNN[7], мають високу точність, вони потребують багато ресурсів. Поєднання розпізнавання з прийняттям рішень дає змогу створювати автономні транспортні системи. Завдяки фреймворкам TensorFlow[8], PyTorch[9] і Keras[10] такі рішення можна швидко реалізувати на практиці.

Початкові методи базувалися на евристичних, що аналізували колір, форму та геометрію знаків. Наприклад, виділення кольору в HSV-просторі, фільтрація контурів і геометричних фігур дозволяли виявляти знаки [11], але були надто чутливими до змін умов освітлення, тіней і часткового перекриття об'єктів.

Подальший розвиток привів до використання алгоритмів машинного навчання з ручним виділенням ознак. Методи на кшталт гістограм орієнтованих градієнтів (HOG) [12], локальних бінарних шаблонів (LBP) [13] і методу головних компонент (PCA) [14] застосовувались для побудови векторних представлень зображень. На їх основі будувались моделі, такі як метод опорних векторів (SVM) [15], метод k-ближчих сусідів (k-NN) [16], наївний байєсівський

класифікатор [17] та Random Forest [18]. Вони забезпечували вищу точність, ніж евристичні підходи, однак вимагали попередньої обробки та були менш гнучкими до нових ситуацій.

Справжній прорив у цій галузі стався з впровадженням глибокого навчання. Згорткові нейронні мережі (CNN) дозволили автоматично виявляти релевантні ознаки зображення без ручного втручання. Вони стали основою для систем класифікації, наприклад ResNet або MobileNet, а також для моделей детекції (YOLO, SSD, Faster R-CNN) і семантичної сегментації, що виділяють форму об'єкта піксельно [19]. Такі моделі демонструють високу точність, стійкість до змін середовища та здатність до генералізації.

Новітнім напрямом є використання трансформерів, зокрема Vision Transformers (ViT) [20], які працюють із зображеннями як із послідовностями токенів, подібно до обробки природної мови. ViT потребує менше обчислювальних ресурсів на етапі тренування та демонструє хорошу ефективність на задачах з обмеженим обсягом даних. Крім того, активно розвиваються гібридні моделі, що об'єднують CNN і трансформери, поєднуючи високу точність із продуктивністю.

Отже, сучасні методи розпізнавання дорожніх знаків пройшли шлях від простих алгоритмів до глибоких і трансформерних моделей, що забезпечують високу точність, гнучкість і масштабованість. CNN довели свою ефективність як оптимальне рішення для побудови інтелектуальних транспортних систем завдяки точності, самонавчанню та адаптивності.

1.3 Аналіз існуючих програмних засобів та наукових рішень

Сучасні системи розпізнавання дорожніх знаків активно впроваджуються у сферу інтелектуального транспорту, автономного керування та безпеки дорожнього руху. Багато автовиробників, наукових установ і стартапів розробили власні рішення для автоматичного виявлення й інтерпретації дорожніх знаків. У цьому розділі розглянуто найпоширеніші та найбільш

ефективні з існуючих систем, проаналізовано їхні особливості, архітектури та сфери застосування.

Системи розпізнавання в складі ADAS. Більшість сучасних автомобілів класу середнього та преміум-сегменту оснащуються системами ADAS (рисунок 1.1). Advanced Driver Assistance Systems електронна система, створена для підтримки водія під час керування автомобілем і процесу паркування. Завдяки інтеграції безпечного людино-машинного інтерфейсу ADAS підвищує рівень безпеки на дорогах і сприяє запобіганню аварій[21].

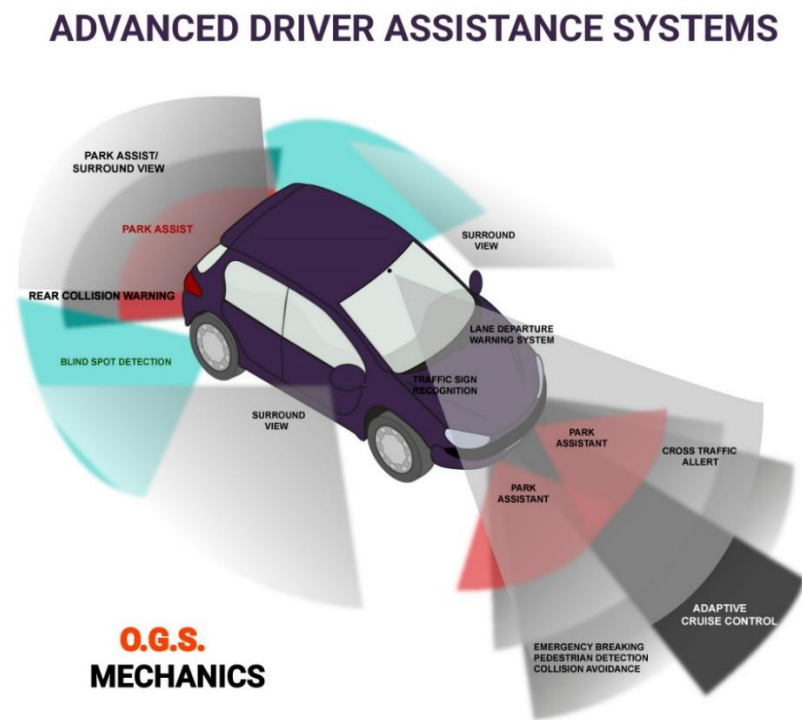


Рисунок 1.1 – система ADAS[22]

У роботі систем ADAS використовуються сучасні автоматизовані технології, такі як датчики та відеокамери, які допомагають виявляти перешкоди або помилки водія, зокрема ознаки його втоми чи сонливості, і забезпечують відповідну реакцію. До переваг даної системи можна віднести:

- Підвищення безпеки дорожнього руху;
- Виявлення знаків у реальному часі;
- Інтеграція з іншими функціями ADAS.

Недоліки:

- Залежність від якості камер та умов освітлення;
- Висока вартість впровадження в бюджетних автомобілях.

Система Mercedes-Benz Traffic Sign Assist (рисунок 1.2) це інноваційна технологія допомоги водієві, яка сприяє покращенню безпеки та комфорту, забезпечуючи розпізнавання та візуалізацію дорожніх знаків у реальному часі. Система розпізнає знаки обмеження швидкості та заборони обгону, відображаючи інформацію на панелі приладів. В основі – глибокі нейронні мережі, які працюють у зв'язці з навігаційною інформацією[23].

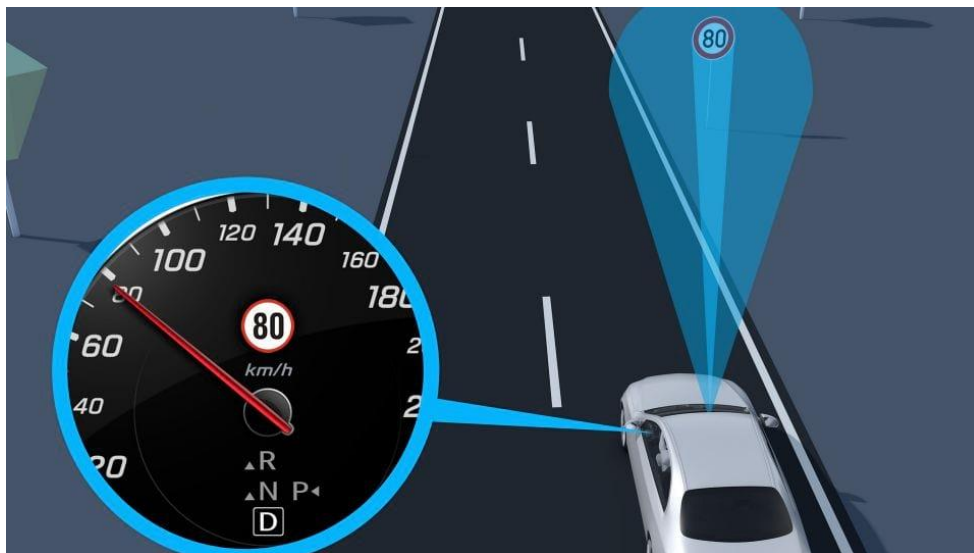


Рисунок 1.2 – система Mercedes-Benz Traffic Sign Assist розпізнавання обмежень швидкості[24]

Перевагами даної системи є:

- Висока точність розпізнавання;
- Зручна візуалізація інформації для водія.

Недоліки:

- Обмежена доступність у базових моделях;
- Менш ефективна при несприятливих погодних умовах.

Система Tesla Autopilot (рисунок 1.3), впроваджує багатокamerну систему, яка працює разом із вдосконаленими нейронними мережами для

точного розпізнавання та класифікації об'єктів у режимі реального часу. Це дозволяє автомобілям визначати дорожні знаки, розпізнавати світлофори, помічати пішоходів і аналізувати інші важливі елементи дороги, що забезпечує високий рівень безпеки[25]. Переваги:

- Постійне вдосконалення через колективне навчання;
- Широкий спектр розпізнаваних об'єктів.

Недоліки:

- Висока потреба у великих обсягах даних;
- Часто вимагає стабільного інтернет-з'єднання для оновлень.



Рисунок 1.3 – система Tesla Autopilot[26]

Окрім комерційних рішень, активно ведуться дослідження у науковому середовищі. Прикладами таких розробок є Tsinghua-Tencent 100K (рисунок 1.4). Це масштабний набір даних, призначений для розпізнавання та класифікації дорожніх знаків у реальних умовах. Він містить 100 000 зображень, які були зібрані з панорам Tencent Street View, серед яких 30 000 є дорожніми знаками. Особливістю цього набору є багатогранність: знімки охоплюють різноманітні погодні умови та рівні освітлення, що дозволяє перевіряти точність моделей у реальних сценаріях[27].

Переваги:

- Універсальність і масштабність;
- Ідеальний для тестування алгоритмів комп'ютерного зору.

Недоліки:

- Не є самостійною системою, а лише основа для досліджень;
- Потребує глибокої технічної обробки.

Інтеграція таких даних значно сприяє покращенню роботи алгоритмів комп'ютерного зору, зокрема згорткових нейронних мереж.

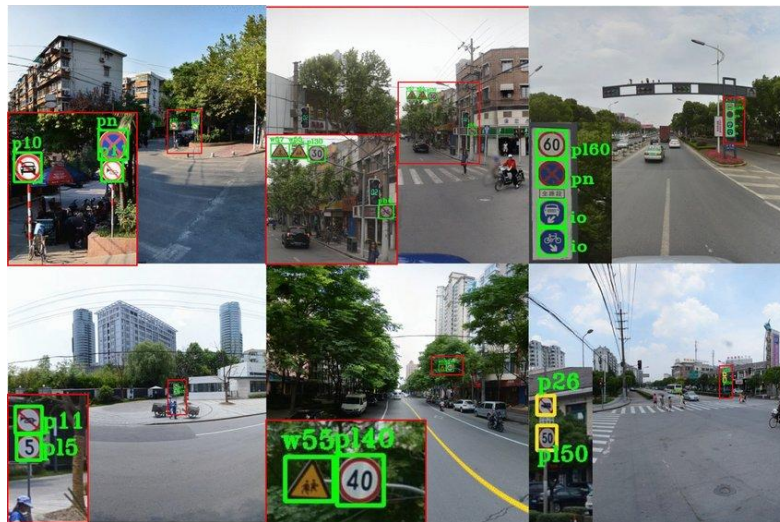


Рисунок 1.4 – система Tsinghua-Tencent 100K[28]

Також не слід забувати про мобільні додатки. Одним з таких є Sygic GPS Navigation & Maps (рисунок 1.5) – це сучасний додаток для навігації, який пропонує користувачам зручність використання офлайн– карт, які регулярно оновлюються щомісяця. Ця програма забезпечує точні попередження про ситуації на дорогах, а також повідомляє про наявність камер контролю швидкості[29].

Переваги:

- Доступність для широкого кола користувачів;
- Можливість використання офлайн.

Недоліки:

- Нижча точність порівняно з вбудованими системами;
- Обмежений функціонал.

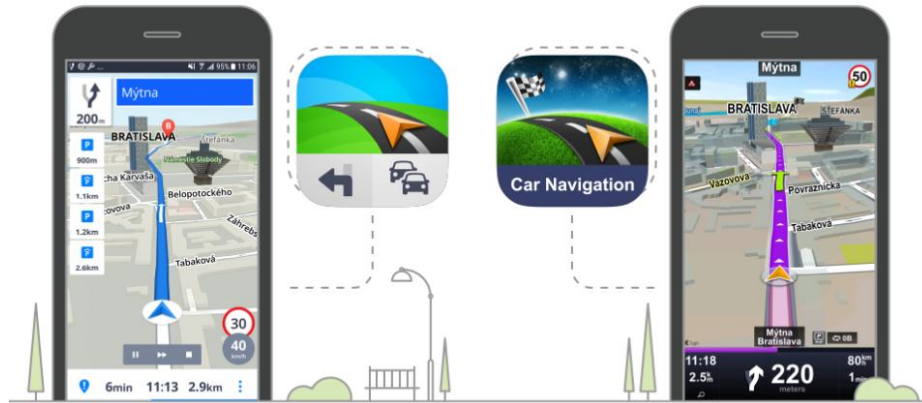


Рисунок 1.5 – Sygic GPS Navigation & Maps[30]

Отже, сучасні системи розпізнавання дорожніх знаків ефективно працюють завдяки глибокому навчанню та розвитку комп'ютерного зору. Комерційні рішення, такі як Mercedes-Benz Traffic Sign Assist і Tesla Autopilot, забезпечують точне розпізнавання об'єктів у реальному часі. Наукові дослідження й набори даних, як Tsinghua-Tencent 100K, слугують основою для вдосконалення таких технологій. Мобільні додатки роблять ці функції доступними для ширшої аудиторії. Усі ці рішення сприяють підвищенню безпеки на дорогах і розвитку розумного транспорту.

1.5 Мета, задачі та вимоги до реалізації інформаційної системи

Таким чином метою КРБ є підвищення точності розпізнавання дорожніх знаків шляхом проєктування методу на основі згорткової нейронної мережі ResNet50.

Для досягнення цієї мети ставляться наступні задачі:

- Виконати аналіз наявних методів розпізнавання дорожніх знаків на основі обробки їхніх зображень, оцінюючи їхні переваги та недоліки.
- Дослідити можливості та обмеження застосування моделей глибокого навчання для ідентифікації дорожніх знаків з урахуванням різноманітних умов та викликів.

– Розробити метод для автоматичного розпізнавання дорожніх знаків, використовуючи сучасні методи глибокого навчання.

– Провести тестування програмної реалізації методу на різноманітних наборах даних.

Результатом має стати працездатна модель, здатна ефективно розпізнавати дорожні знаки у різних умовах.

1.6 Висновки до розділу 1

У розділі було здійснено комплексний аналіз предметної області, що стосується задачі розпізнавання дорожніх знаків. Розглянуто еволюцію підходів – від традиційних алгоритмів до глибоких згорткових нейронних мереж, які продемонстрували найвищу ефективність у складних реальних умовах. Оцінено сучасні архітектури CNN, такі як YOLO, ResNet, MobileNet, а також перспективи використання Vision Transformers.

Проведений огляд програмних рішень, серед яких системи ADAS, Tesla Autopilot, Mercedes-Benz Traffic Sign Assist, наукові набори даних та мобільні додатки, показав різноманітність інструментів і підтвердив доцільність використання нейронних мереж у побудові систем комп'ютерного зору. Виокремлено переваги та недоліки кожного з підходів.

Таким чином, використання глибокого навчання є найбільш доцільним і перспективним шляхом для реалізації інформаційної системи розпізнавання дорожніх знаків, що дозволить досягти високої точності, адаптивності та стабільності роботи в умовах реального середовища.

Розділ 2 Розроблення методу

2.1 Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання

Щоб дослідити та пояснити процес розпізнавання дорожніх знаків за допомогою технологій глибокого навчання, була розроблена схема методу, яка представлена на рисунку 2.1.

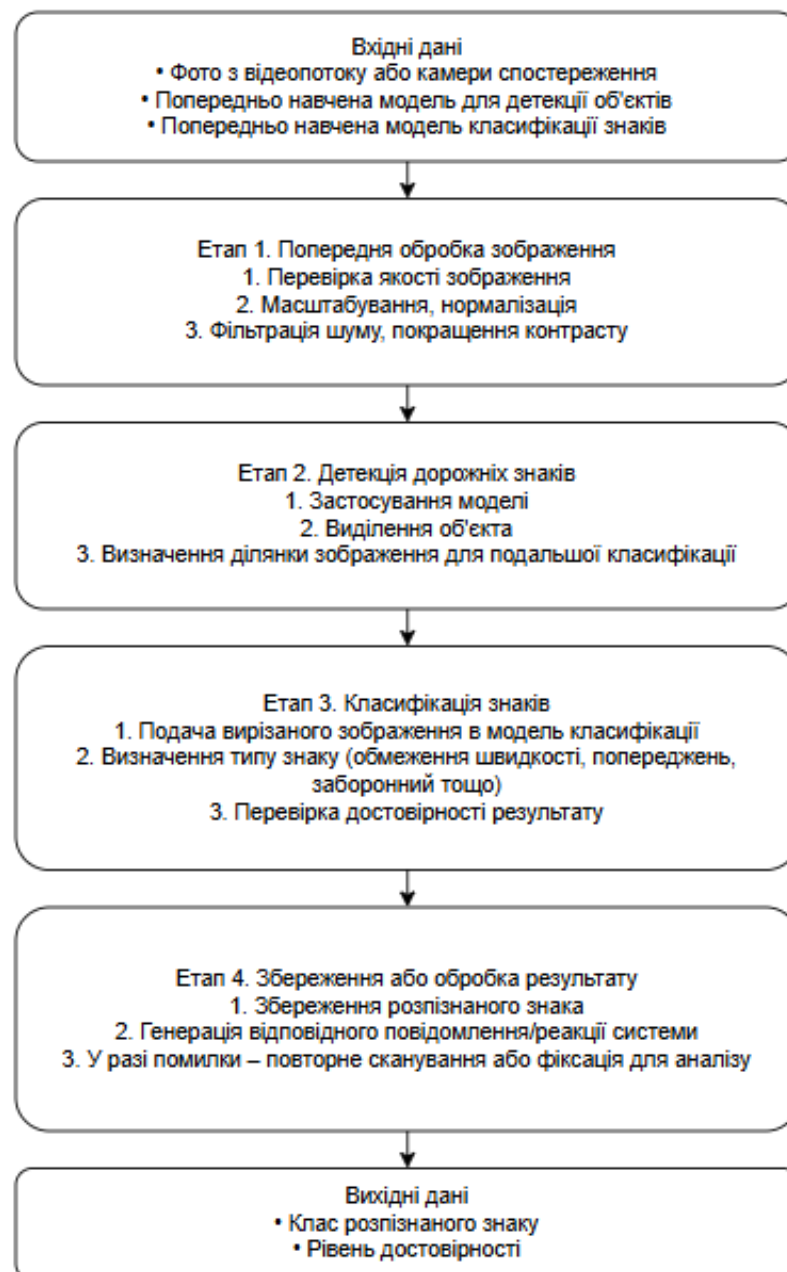


Рисунок 2.1 – Схема методу розпізнавання дорожніх знаків за допомогою моделей глибокого навчання.

Далі наведено детальний опис кожної частини схеми.

Вхідними даними методу автоматичного розпізнавання дорожніх знаків є: Фото з відеопотоку або камери спостереження. Зображення, на якому проводиться пошук дорожніх знаків. Фото може бути отримане з камер спостереження, транспортного засобу або мобільного пристрою.

Файл тренованої моделі для детекції об'єктів. Файл містить параметри попередньо навченої нейронної мережі, здатної виявляти об'єкти на зображеннях, дорожні знаки.

Файл тренованої моделі для класифікації. Це модель, яка аналізує вирізані фрагменти зображень та визначає клас дорожнього знака (обмеження швидкості, заборона руху тощо).

Етап 1 – Попередня обробка зображення

Перевірка зображення, програма перевіряє, чи завантажений файл є дійсним зображенням (розширення, формат, розмір тощо). У випадку невідповідності виводиться повідомлення: «Файл не визначено».

Масштабування та нормалізація, зображення приводиться до стандартного розміру. Нормалізація пікселів дозволяє мережі ефективніше працювати зі значеннями.

Фільтрація шуму, покращення контрасту методами зменшення шумів (фільтри Гауса, медіанний фільтр) для покращення чіткості зображення.

Етап 2 – Детекція дорожніх знаків

Застосування моделі ResNet50/SSD які запускають алгоритм, який аналізує зображення та виявляє ймовірні місця розташування знаків.

Визначення меж знака, для цього встановлюються координати області (bounding box), де виявлено знак.

Вирізання фрагмента із знаком, що вирізають окремий фрагмент зображення, який містить знак, для подальшої класифікації.

У разі, якщо жодного знака не знайдено, виводиться повідомлення: "На фото нема дорожніх знаків".

Етап 3 – Класифікація дорожніх знаків

Передача зображення до класифікатора потім вирізаний фрагмент передається до моделі класифікації на основі CNN.

Розпізнавання типу знака ця модель визначає тип дорожнього знака (наприклад, «СТОП», «50 км/год», «Пішохідний перехід»).

Оцінка достовірності за допомогою якої визначається рівень впевненості системи у правильності розпізнавання. Якщо рівень нижчий за поріг – запускається повторна спроба або повідомлення користувачу.

У разі невдалого розпізнавання виводиться повідомлення: «Не вдалося розпізнати знак на фото».

Етап 4 – Збереження результату та вивід

Запис у базу даних, успішно розпізнаний знак записується до бази (клас, координати, час, достовірність).

Вивід зображення з результатом, що виводить фрагмент фото із розпізнаним знаком, тип знака та його характеристика.

Генерація повідомлення, в системі формується сповіщення або реакція (наприклад, звуковий сигнал чи командна дія), для того щоб водій звернув свою увагу.

Вихідні дані:

Тип розпізнаного знака – текстовий опис, наприклад: «Знак обмеження швидкості 50»

Рівень достовірності – достовірність розпізнавання у відсотках.

Отже, розглянута схема методу автоматичного розпізнавання дорожніх знаків на основі глибокого навчання відображає логічну послідовність обробки зображень від попередньої підготовки та детекції до класифікації та збереження результатів. Деталізований опис кожного етапу дозволяє краще зрозуміти функціонування системи та обґрунтувати вибір технологій, що застосовуються. Запропонована структура є гнучкою і може бути адаптована під різні технічні умови та потреби практичного використання в інтелектуальних транспортних системах.

2.2 Функціональна структура інформаційної системи

У цьому розділі здійснено розгляд принципів організації та внутрішньої логіки функціонування системи, яка реалізує метод розпізнавання дорожніх знаків на основі технологій глибокого навчання. Щоб максимально наочно і послідовно пояснити процес, було виділено основні групи функцій, кожна з яких виконує окрему операцію в межах загального алгоритму роботи системи.

Метод, реалізований у структурі, складається з шести логічних блоків функціональності:

1. Завантаження і ініціалізація моделей нейронних мереж.
2. Обробка вхідного зображення.
3. Виявлення дорожніх знаків на зображенні.
4. Виокремлення знайдених об'єктів та приведення їх до стандартного вигляду.
5. Ідентифікація типу дорожнього знаку.

Нижче розглянемо кожен із зазначених етапів більш докладно.

Ініціалізація моделей нейронних мереж – на цьому етапі виконується завантаження моделей, попередньо навченої на задачах детекції (YOLO або SSD) та класифікації (CNN, MobileNet). Також задаються необхідні параметри для запуску – вибір класів, шляхи до файлів ваг, конфігурацій, формат вхідних даних.

Обробка зображення – основна мета цього блоку – оптимізація зображення для подальшої роботи з моделлю. Передбачено масштабування до фіксованого розміру, нормалізацію яскравості та контрасту, зменшення шумів, а також, за потреби, перетворення до градацій сірого кольору для спрощення обчислень.

Детекція об'єктів – цей блок здійснює аналіз зображення з метою локалізації потенційних дорожніх знаків. Алгоритм детекції генерує прямокутні області (bounding boxes), які обмежують знайдені об'єкти. Якщо ознак дорожніх знаків не виявлено, програма фіксує відповідне повідомлення.

Вирізання та стандартизація зображень знаків – об’єкти, визначені у попередньому етапі, вирізаються та обробляються геометрично для уніфікації – зокрема, виконується масштабування та, при потребі, корекція перспективи (афінні перетворення). Це забезпечує узгодженість перед подачею в класифікатор.

Класифікація знаків – підготовлені зображення подаються до згорткової нейронної мережі, яка визначає, до якого типу належить знак: попереджувальний, обмежувальний, сервісний тощо. Крім класу, модель формує показник ймовірності (confidence score), який вказує на точність передбачення. При недостатній достовірності система може ініціювати повторну перевірку.

Реєстрація результату – після успішного розпізнавання інформація про знак (тип, координати, час фіксації, рівень впевненості) зберігається у структурованому форматі. Дані можуть використовуватись для візуалізації, звітності або керування іншими модулями –наприклад, ADAS-системами.

Таким чином, функціональна модель системи розпізнавання дорожніх знаків передбачає логічну побудову з послідовними етапами, кожен із яких має чітке завдання. Такий підхід дозволяє забезпечити ефективну, надійну і масштабовану роботу алгоритму, що є особливо важливим при інтеграції у реальні дорожні умови або системи автономного керування.

2.3 Архітектура нейронної мережі

Для реалізації методу автоматичного розпізнавання дорожніх знаків із зображень використовується поєднання інструментів комп’ютерного зору (OpenCV) та глибокого навчання (TensorFlow або PyTorch). У цьому підрозділі детально описано архітектуру згорткової нейронної мережі (CNN) та допоміжні алгоритми обробки зображень, які застосовуються для виділення та класифікації дорожніх знаків.

Використання OpenCV для попередньої обробки зображення. OpenCV – це відкрита бібліотека комп’ютерного зору, яка надає інструменти для роботи з графікою, зображеннями та відео в реальному часі. На початковому етапі вона використовується для обробки зображення перед подачею до нейромережі[31].

Архітектура попередньої обробки передбачає такі компоненти: вхідні дані, попередня обробка зображення, виділення області знака та вихідні дані.

Вхідні дані:

- Формат зображення RGB або Grayscale.
- Розмір довільний в наслідок адаптації до стандартного.
- Джерело зображення камера або відеофайл.

Попередня обробка зображення:

- Перетворення до градацій сірого.
- Застосування фільтра Гауса для зменшення шумів.
- Нормалізація пікселів до діапазону [0,1].
- Масштабування до розміру, прийнятного для моделі.

Виділення області знака, робиться для спрощеної реалізації можливе застосування попереднього фільтрування за кольором або контуром. У повній реалізації, використання моделі ResNet50 для детекції об’єкта (bounding box).

Вихідні дані: координати виявленого знака (x, y, ширина, висота).

Архітектура згорткової нейронної мережі для класифікації дорожніх знаків здійснюється за допомогою згорткової нейронної мережі (CNN), розробленої спеціально для обробки зображень. Архітектура складається з таких компонентів: вхідний шар, згортковий шар, шари нормалізації, повнозв’язні шари та вихідний шар.

Вхідний шар:

- Розмірність: $64 \times 64 \times 3$ (кольорове зображення).
- Завдання: передача зображення до моделі.

Згорткові шари:

- 2–3 згорткових шару з ядрами 3×3 .
- Кількість фільтрів: 32, 64, 128.

- Функція активації: ReLU.
- Після кожного згорткового шару – шар підвибірки (MaxPooling 2×2).

Шари нормалізації та Dropout:

- Застосовуються для стабілізації та запобігання перенавчанню.
- Dropout на рівні 0.25–0.5.

Повнозв'язні шари:

- Перетворення плоского вектору ознак у класову розмітку.
- Softmax – для класифікації по 10–43 класах (в залежності від типу датасету).

Вихідний шар:

- Вивід класу знака (наприклад, «СТОП», «Обмеження 50», «Пішохідний перехід»).
- Confidence Score (0–100%)[32].
- Розмірність вхідних і вихідних даних.

Вхідні дані: зображення дорожнього знака 64×64 пікселів, 3 канали (RGB).

Вихідні дані:

- Назва класу дорожнього знака.
- Ймовірність класифікації.
- Координати об'єкта на фото.

Інтеграція компонентів у загальну систему, що функціонує таким чином:

- Вхідний шар приймає зображення, виконує його масштабування і перед обробку.
- Блок детекції (OpenCV або YOLO) виявляє можливі зони знаків.
- Зони подаються на вхід нейронної мережі, яка класифікує знаки.
- Результати передаються у вигляді тексту або зберігаються в базу даних.
- Інтерфейс відображає вхідне фото з підписом знайденого знака.

Розроблена архітектура забезпечує надзвичайно високу точність, зберігаючи при цьому високий рівень швидкодії. Це особливо важливо для

використання в умовах реального дорожнього середовища та інтеграції у мобільні системи, де оптимальна продуктивність є критичною.

2.4 Проектування архітектури інтелектуальної системи та взаємозв'язок компонентів

На основі методу розпізнавання дорожніх знаків за допомогою моделей глибокого навчання була розроблена архітектура системи, яка представлена на рисунку 2.2. Вона включає наступні компоненти:

- підсистема обробки зображень;
- підсистема управління базою даних;
- підсистема розпізнавання дорожніх знаків.

Підсистема аналізу зображень виконує такі ключові завдання:

- здійснює прийом вхідних зображень, отриманих із відеокамер або інших пристроїв захоплення візуальної інформації;
- проводить первинну обробку отриманих зображень, що включає покращення якості та зниження рівня візуального шуму;
- реалізує процедуру виявлення контурів та визначення важливих ділянок зображення, в яких можуть знаходитися номерні знаки;
- забезпечує аналіз просторових характеристик номерних знаків, зокрема визначення їх положення, розміру та орієнтації з метою подальшої ідентифікації.

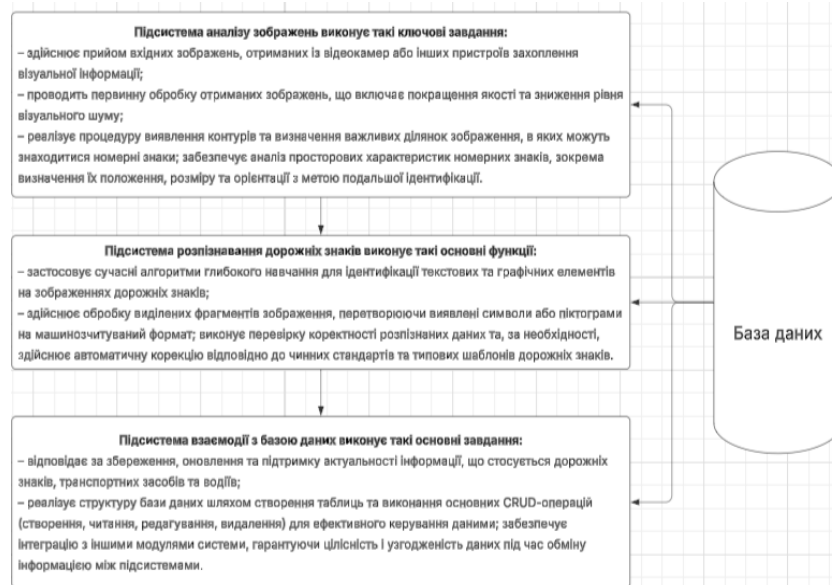


Рисунок 2.2 – Схема архітектури системи розпізнавання дорожніх знаків

Підсистема розпізнавання дорожніх знаків виконує такі основні функції:

- застосовує сучасні алгоритми глибокого навчання для ідентифікації текстових та графічних елементів на зображеннях дорожніх знаків;
- здійснює обробку виділених фрагментів зображення, перетворюючи виявлені символи або піктограми на машинозчитуваний формат;
- виконує перевірку коректності розпізнаних даних та, за необхідності, здійснює автоматичну корекцію відповідно до чинних стандартів та типових шаблонів дорожніх знаків.

Підсистема взаємодії з базою даних виконує такі основні завдання:

- відповідає за збереження, оновлення та підтримку актуальності інформації, що стосується дорожніх знаків, транспортних засобів та водіїв;
- реалізує структуру бази даних шляхом створення таблиць та виконання основних CRUD-операцій (створення, читання, редагування, видалення) для ефективного керування даними;
- забезпечує інтеграцію з іншими модулями системи, гарантуючи цілісність і узгодженість даних під час обміну інформацією між підсистемами.

Діаграма варіантів використання відображає основні сценарії роботи з програмою та дозволяє показати ключові дії, які може виконувати користувач. На рисунку 2.3 представлено графічне зображення цієї діаграми. Вона включає

такі дії, як запуск програми, обробка відео потоку з камери, розпізнавання дорожніх знаків, відображення результатів класифікації та завершення сеансу. Кожен варіант використання ілюструє окремий функціональний елемент, з яким взаємодіє користувач у межах системи.

Для полегшення розуміння логіки роботи програмного забезпечення, під кожним сценарієм надається стислий опис його призначення. Це дозволяє наочно уявити, як реалізується робота з додатком у реальних умовах, а також полегшує подальший аналіз або вдосконалення системи.

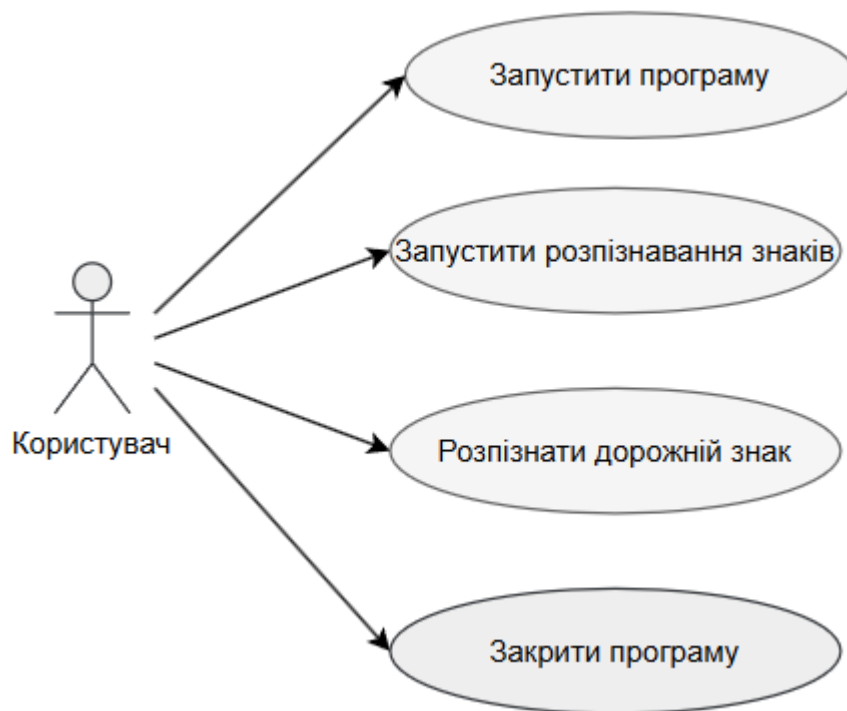


Рисунок 2.3 – Діаграма варіантів використання

Діаграма варіантів використання показує, як саме користувач взаємодіє з програмою для розпізнавання дорожніх знаків:

– Спочатку користувач запускає програму. Це стандартна дія, яка дозволяє відкрити програму й підготувати її до роботи. В цей момент відбувається завантаження всіх необхідних компонентів, і система готова до подальших дій.

– Після запуску користувач переходить до наступного кроку — вмикає розпізнавання знаків. Це означає, що активується відповідний модуль, який

починає роботу з камерою або зображеннями. Програма готується обробляти дані й шукати на них дорожні знаки.

– Далі відбувається сам процес розпізнавання. Коли користувач надає зображення або відео, програма аналізує його і намагається визначити, який саме знак зображено. Для цього використовується модель глибокого навчання, яка була попередньо натренована на прикладах. Результат розпізнавання показується на екрані.

– На завершення користувач закриває програму. Це звичайна дія, яка дозволяє вийти з системи, зупинити всі процеси і звільнити ресурси. Програма завершує свою роботу.

Ці функціональні можливості забезпечують повний цикл роботи програми, починаючи від запуску та розпізнавання знаків, і закінчуючи закриттям програми, що робить програму зручною та ефективною для користувача.

2.5 Підготовка датасету для нейромережевого розпізнавання дорожніх знаків

Якість класифікації безпосередньо залежить від якості та структури навчальних даних. Для задачі автоматичного розпізнавання дорожніх знаків необхідно підготувати репрезентативний датасет, який включає зображення різних типів знаків у різних умовах освітлення, ракурсах та фонових завадах.

Формування навчального датасету для системи розпізнавання дорожніх знаків здійснюється з декількох джерел:

Публічні датасети, використовуються відкриті та загальнодоступні набори зображень дорожніх знаків, такі як GTSRB та BelgiumTSC. Вони містять тисячі прикладів різних типів знаків, що дозволяє ефективно тренувати нейромережу та проводити об'єктивну перевірку її точності на стандартних тестових наборах.

Власні фотографії, до навчальної вибірки включено зображення, зроблені самостійно в реальних умовах –на дорогах, у різних погодних ситуаціях, при

змінному освітленні. Ці дані особливо цінні для адаптації моделі до національних особливостей дорожніх знаків та локальних сценаріїв.

Комбіновані джерела, з метою покращення узагальнюючої здатності моделі, навчальний датасет було сформовано шляхом об'єднання публічних та власноруч зібраних зображень. Такий підхід дозволяє підвищити різноманітність прикладів і забезпечити стабільність моделі при роботі в умовах, відмінних від ідеальних.

Для навчання та тестування моделі глибокого навчання з розпізнавання дорожніх знаків використовуються комбіновані джерела даних. Це дозволить досягти балансу між кількістю прикладів, різноманітністю класів та якістю зображень.

GTSRB (German Traffic Sign Recognition Benchmark) – один із найпопулярніших наборів зображень дорожніх знаків, що містить понад 50 000 зображень 43 класів[33].

BelgiumTSC – додатковий датасет із реалістичними умовами зйомки, що використовується для тестування моделі на складних зображеннях[34].

Власноруч зібрані дані – фотографії дорожніх знаків, зроблені у місцевих умовах для підвищення адаптивності моделі. Оскільки відкриті датасети здебільшого сформовані на прикладі дорожніх знаків європейських країн, виникає потреба у локалізації навчального матеріалу. Для цього було зібрано власний набір фотографій знаків українських доріг.

Об'єднання візуальних даних із кількох джерел дозволяє створити багатий і всебічний набір інформації, який відіграє вирішальну роль у процесі навчання та тестування програмного продукту. Такий підхід забезпечує всебічну підготовку та перевірку програмного забезпечення для роботи в різноманітних умовах і середовищах.

2.6 Особливості використання спеціалізованих програмних компонентів

Розробка системи розпізнавання дорожніх знаків на основі глибокого навчання неможлива без застосування сучасних програмних інструментів і бібліотек. Ці компоненти забезпечують ефективну обробку зображень, тренування нейронних мереж, візуалізацію результатів та інтеграцію із зовнішніми пристроями.

Одним із ключових інструментів є бібліотека OpenCV[35], яка широко використовується в галузі комп'ютерного зору. OpenCV застосовується для попередньої обробки зображень: масштабування, зменшення шуму, перетворення кольорових моделей та виділення контурів.

Для побудови та навчання нейронної мережі використовуються середовища TensorFlow або PyTorch. Обидві ці платформи дозволяють створювати складні згорткові моделі (CNN), застосовувати сучасні оптимізатори, контролювати процес навчання, а також реалізовувати GPU-прискорення, що значно знижує час обробки великих обсягів даних.

У проєкті застосовуються фреймворк Keras, який надає високорівневий інтерфейс для TensorFlow. Він дозволяє швидко проєктувати архітектуру нейронної мережі, компілювати її та запускати навчання з моніторингом точності та функції втрат. Інтеграція з Matplotlib[36] та TensorBoard[37] забезпечує зручну візуалізацію навчального процесу.

Для розмітки навчального набору зображень використовуються програми LabelImg та Roboflow[39]. Перша дозволяє вручну позначати області дорожніх знаків на зображеннях, створюючи відповідні анотації. Друга є хмарною платформою, що дозволяє виконувати як розмітку, так і аугментацію зображень, а також експортувати їх у необхідному форматі –YOLO, COCO чи Pascal VOC[39].

На етапі аналізу результатів та виведення метрик були використані бібліотеки Matplotlib, Seaborn та Pandas. Вони застосовуються для побудови графіків навчання, оцінки точності, візуалізації розподілу класів та створення таблиць із результатами класифікації.

Таким чином, використання спеціалізованих розширень дозволить створити повноцінну систему розпізнавання дорожніх знаків із високим рівнем функціональності, що легко адаптується до різних технічних умов та програмних середовищ.

2.7 Оцінювання точності розпізнавання дорожніх знаків

На етапі оцінювання методу розпізнавання дорожніх знаків необхідно застосовувати низку метрик, які дозволяють комплексно проаналізувати якість класифікації. Зважаючи на те, що розроблена модель працює з багато класовою задачею, оцінювання має бути проведене як на загальному, так і на класовому рівні.

Насамперед використовується точність класифікації (accuracy), яка визначає частку зображень, правильно віднесених до свого класу. Вона дає загальне уявлення про ефективність моделі, проте не завжди є інформативною у випадках нерівномірного розподілу класів у вибірці.

Для більш глибокого аналізу використовуються прецизійність (precision) та повнота (recall) для кожного класу. Прецизійність показує, наскільки модель точно визначає клас при передбаченні, а повнота – наскільки повно вона знаходить усі зображення певного класу. Для об'єднання цих двох показників використовується F1-міра, яка дозволяє оцінити баланс між точністю та повнотою.

Ще одним важливим інструментом є матриця помилок (confusion matrix), яка демонструє, які саме класи найчастіше плутає модель. Це дозволяє виявити типові помилки класифікації та слабкі місця у роботі системи.

У багатокласових задачах також корисно застосовувати усереднені метрики, зокрема:

– макро-середнє (macro averaging) – коли розрахунки проводяться по кожному класу незалежно, а потім усереднюються;

– зважене середнє (weighted averaging) – коли враховується кількість зразків у кожному класі.

Крім точності, важливим критерієм для практичного застосування є швидкість класифікації. Для цього вимірюється середній час обробки одного зображення, що дозволяє оцінити, наскільки модель підходить для роботи в режимі реального часу.

Також проводиться аналіз помилок класифікації. Неправильно розпізнані зображення групуються за типом помилок для виявлення основних причин – наприклад, через низьку якість зображення, нехарактерний вигляд знака або недостатню кількість навчальних прикладів певної категорії.

Оцінка точності, аналіз результатів і тестування на змінених зображеннях (зі змінами контрасту, шумами чи обертанням) дозволяє не лише перевірити надійність моделі, але й визначити напрямки для її подальшого вдосконалення.

2.8 Висновки до розділу 2

З цього розділу було докладно розкрито метод виявлення дорожніх знаків із застосуванням моделей глибокого навчання. Було розроблено і детально описано алгоритм, що охоплює всі етапи обробки вхідного зображення: від попередньої підготовки даних до збереження результатів розпізнавання. Запропонована методика базується на сучасних згорткових нейронних мережах та демонструє логічну та функціонально узгоджену структуру.

Велику увагу приділено формуванню навчального набору, до якого увійшли зображення з відкритих і локальних джерел, а також виконано аугментацію для покращення якості узагальнення моделі.

Окремо розглянуто використання спеціалізованих інструментів, таких як OpenCV, PyTorch, Roboflow і LabelImg, які забезпечують повний цикл розробки — від підготовки даних і розмітки до навчання нейронної мережі та виводу результатів.

На завершення розділу було описано метрики оцінювання точності роботи моделі: точність (accuracy), прецизійність (precision), повнота (recall), F1-міра, а також матриця помилок. Використання цих показників дозволить надалі не лише об'єктивно перевірити якість класифікації, а й виявити слабкі місця для подальшого вдосконалення системи.

У результаті в цьому розділі сформовано технічну та методологічну базу, необхідну для подальшої реалізації та тестування системи розпізнавання дорожніх знаків.

Розділ 3 Експериментальне дослідження методу та реалізація програмного застосунку

3.1 Визначення шляхів дослідження та засобів створення програмного застосунку

Основною метою дослідження є перевірка гіпотези про те, що використання глибокого навчання дозволяє суттєво підвищити точність розпізнавання дорожніх знаків у режимі реального часу. Для реалізації цієї мети обрано шлях створення експериментального програмного застосунку, який поєднує інструменти комп'ютерного зору та штучного інтелекту.

Розроблений програмний застосунок дозволяє виконувати наступні дії:

- обробку зображень, отриманих із відео потоку веб камери або завантажених файлів;
- виявлення областей, що потенційно містять дорожні знаки;
- передачу зображення знака у згорткову нейронну мережу;
- визначення класу знака на основі навченої моделі;
- візуалізацію результату класифікації у зручному для користувача форматі.

Нижче представлено п'ять основних кроків, які дозволили встановити оптимальні шляхи реалізації.

Крок 1: Формування навчального і тестового наборів, для ефективного навчання моделі розпізнавання дорожніх знаків було використано відкритий набір даних German Traffic Sign Recognition Benchmark (GTSRB), який містить понад 50 000 зображень, розподілених по 43 категоріях.

Крок 2: Вибір технологій, на початковому етапі реалізації проєкту було здійснено аналіз актуального технологічного стеку, придатного для розробки системи автоматичного розпізнавання дорожніх знаків. Визначалась цільова платформа, мова програмування, середовище розробки, текстовий редактор. Також були обрані інструменти для адміністрування бази даних та модулі виводу результатів.

Крок 3: Розробка програмного застосунку. На цьому етапі реалізовано основну функціональність системи – від обробки зображень до класифікації дорожніх знаків. Було створено модуль нейронної мережі, який приймає зображення, виконує попередню обробку та класифікацію.

Крок 4: Тестування системи, для перевірки правильності роботи системи було заплановано проведення повного циклу тестування, включаючи модульне, інтеграційне та системне. Модульне тестування проводилося в процесі написання кожного окремого компонента. Після об'єднання модулів у цілісну систему було проведено інтеграційне тестування з метою перевірки сумісності та взаємодії між компонентами.

Крок 5: Дослідження, для оцінки ефективності запропонованого методу було проведено серію експериментів, що охоплюють аналіз точності розпізнавання дорожніх знаків в різних умовах, включаючи зміну освітлення, ракурсу та якості зображень. Крім того, виконано порівняння продуктивності різних моделей глибокого навчання, а також оцінено вплив розміру та якості датасету на результати класифікації. У кожному випадку визначено критерії точності, зібрано статистичні дані та проведено оцінку результатів за допомогою основних метрик, таких як точність, повнота та F1-міра.

3.2 Засоби розробки програмного застосунку

Розробка програмного застосунку для розпізнавання дорожніх знаків вимагала ретельного підбору програмних засобів, які забезпечують ефективну реалізацію усіх ключових етапів – від попередньої обробки зображень до виводу результатів класифікації. До складу програмного середовища увійшли мова програмування, фреймворк глибокого навчання, бібліотека з попередньо натренованими моделями, інструменти розробки GUI, а також датасет для навчання та тестування.

Основною мовою програмування обрано Python[39] – мову високого рівня з широким набором бібліотек для наукових обчислень, комп'ютерного зору та

штучного інтелекту. Python є ідеальним вибором для задач розпізнавання зображень, зокрема завдяки бібліотекам OpenCV, TensorFlow, Keras, NumPy, Matplotlib та іншим.

Переваги Python у контексті проекту:

- Легка інтеграція з модулями машинного навчання;
- Простота синтаксису, що дозволяє зосередитися на логіці, а не на технічних деталях;
- Вбудовані засоби роботи з файлами, графікою, форматами зображень;
- Велика кількість документації та активна спільнота.

У якості основного інструмента реалізації моделі глибокого навчання використано PyTorch[40] –потужний фреймворк для побудови та навчання нейронних мереж. PyTorch забезпечує високу продуктивність при роботі на CPU та GPU, підтримує автоматичне обчислення градієнтів, а також має активну спільноту з великою кількістю прикладів.

Для реалізації використано нейронну мережу ResNet50[41], що відзначається глибокою архітектурою та здатністю до ефективного аналізу візуальних патернів. Бібліотека torchvision забезпечила доступ до попередньо навчених ваг, що суттєво зменшило час навчання та дозволило досягти високої продуктивності.

Оскільки початковий формат моделі був призначений для загальних задач класифікації зображень, її було адаптовано до розпізнавання дорожніх знаків за допомогою заміни вихідного шару та повторного навчання на основі датасету GTSRB. Така модифікація сприяла точному розподілу об'єктів у 43 категорії та покращенню узагальнення моделі.

Для ізоляції середовища розробки було створено віртуальне середовище за допомогою `venv`, що дозволило уникнути конфліктів залежностей між різними проектами. Завдяки використанню `venv` кожен компонент залишався у межах локального середовища. Це рішення забезпечило максимальну гнучкість у тестуванні та вдосконаленні алгоритму без ризику впливу зовнішніх змін у загальному середовищі[42].

3.3 Структура та функціональне призначення програмних складових системи

Розроблена система розпізнавання дорожніх знаків базується на модульному підході з чітким розмежуванням функцій між окремими логічними блоками. Програмна система складається з двох основних частин: модуль навчання моделі та модуль розпізнавання знаків у режимі реального часу. Обидві частини працюють автономно та передбачають ізольовану логіку виконання.

Загальна структура системи програмного застосунку складається з таких основних модулів:

- «train_model.py» – навчання та збереження моделі на базі ResNet50;
- «predict_realtime.py» – виведення розпізнаних знаків із камери в реальному часі;
- «sign_labels.csv» – таблиця відповідностей класів і назв дорожніх знаків;
- «myData/myData/» – навчальна вибірка зображень для класифікації;
- «resnet50_traffic.pt» – збережена модель після навчання;
- «arial.ttf» – шрифт для виводу результатів кирилицею (українською мовою).

Модуль навчання train_model.py, цей скрипт відповідає за побудову нейронної мережі ResNet50, її адаптацію до 43 класів дорожніх знаків із датасету GTSRB, а також навчання та збереження результату. Основні складові:

- Ініціалізація параметрів: шлях до датасету, кількість епох, розмір зображення, розмір пакету, вибір GPU/CPU.
- Обробка даних: зображення зчитуються через torchvision.datasets.ImageFolder, трансформуються (Resize, Normalize) і передаються в DataLoader.
- Архітектура: використовується models.resnet50 (weights=ResNet50_Weights.DEFAULT), у якій вихідний шар (fc) замінено на лінійний для 43 класів.

– Навчання: проходить у циклі по епохах, з обчисленням втрат та точності.

– Збереження: модель зберігається у файл `resnet50_traffic.pt`.

Функції:

– `train()` –головна функція тренування;

– `evaluate()` –обчислення точності на валідаційній вибір

Модуль розпізнавання `predict_realtime.py`, цей модуль забезпечує розпізнавання знаків у потоці відео з веб камери з використанням натренованої моделі. Він інтегрує PyTorch, OpenCV і PIL для одночасної обробки та візуалізації результатів.

Основні компоненти:

– Завантаження моделі: модель ResNet50 завантажується з `.pt` файлу та переходить у режим `eval`.

– Трансформація: нове зображення з відео перетворюється у тензор відповідно до тренувальних трансформацій.

– Розпізнавання: застосовується Softmax для отримання ймовірностей та визначення найбільш ймовірного класу.

– Вивід: назва знака виводиться на відеопотік за допомогою PIL та OpenCV, підтримується український текст через `ImageFont.truetype`.

За дані і класова розмітка а також назви знаків та класів відповідає `sign_labels.csv` це таблиця з відповідністю між числовими класами (`folder`) та назвами знаків (`name`). Ці назви відображаються користувачеві після розпізнавання. `ImageFolder` це структура підпапок у `myData/myData`, де кожна папка містить зображення одного класу (назва папки –номер класу).

Передача даних між модулями відбувається через `transform` зображення в обох модулях приводиться до формату тензора ($3 \times 224 \times 224$), нормалізується та подається на вхід ResNet50. Вихідний вектор ймовірностей `softmax` обробляється для знаходження класу `argmax` та рівня достовірності `confidence`, після чого отримані дані перетворюються в текст через `label_dict` і передаються в графічний інтерфейс користувача GUI. Єдиним об'єктом обміну між модулями

`train_model.py` і `predict_realtime.py` є файли моделі `resnet50_traffic.pt`, який зберігається після навчання та завантажується під час розпізнавання.

3.4 Особливості реалізації програмних складових системи

Процес програмної реалізації системи розпізнавання дорожніх знаків охоплює кілька ключових компонентів, серед яких: підготовка даних, побудова та навчання моделі, інтеграція результатів класифікації в графічний інтерфейс, а також обробка відеопотоку в режимі реального часу. Реалізація кожного з цих етапів супроводжувалася вирішенням окремих інженерних завдань.

Підготовка та обробка вхідних зображень. Перед подачею на вхід моделі всі зображення проходять через трансформаційний ланцюг: зміна розміру до 224×224 пікселів, нормалізація каналів кольору відповідно до статистики ImageNet, а також перетворення у тензори. Ці дії реалізовані через конвеєр `transforms.Compose`, що забезпечує однакову структуру даних як під час навчання, так і в реальному використанні.

Особливістю є підтримка розпізнавання зображень у різних форматах (`.jpg`, `.png`) та з різним співвідношенням сторін. Для збереження геометричної цілісності було використано централізоване масштабування із збереженням пропорцій.

Реалізація та адаптація моделі. Модель ResNet50 обрана як основна архітектура для класифікації через її стійкість до перенавчання та здатність обробляти складні візуальні патерни. Модель була завантажена з попередньо навченими вагами (`pretrained=True`), після чого останній шар замінено на лінійний, що адаптований під 43 класи GTSRB. На рисунку 3.1 наведені приклади зображень з комбінованого набору даних, які містять специфікацію класів дорожніх знаків для їх розпізнавання.

Особливість реалізації полягала в тому, що для зменшення обсягу обчислень не проводилося повне перенавчання всієї мережі, а лише `fine-tuning` останніх шарів. Це дозволило зекономити час тренування без втрати точності.



Рисунок 3.1 – Приклади зображень у комбінованому наборі даних показують класи дорожніх знаків для розпізнавання.

Процес тренування моделі зображень (рисунок 3.2), навчання відбувається протягом заданої кількості епох. Кожна епоха включає:

- Проходження всіх навчальних даних: Мережа аналізує кожне зображення, обчислює передбачення та порівнює їх із правильними мітками.
- Розрахунок функції втрат: Визначається рівень помилки моделі та коригуються ваги для її покращення.
- Оновлення параметрів: Використовується зворотне поширення помилки для коригування ваги нейронів.
- Оцінка точності: В кінці кожної епохи підраховується загальна точність на тестових даних.
- Збереження найкращої моделі: Параметри моделі зберігаються у файл .pt, щоб можна було повторно використовувати результат.

Особливістю є використання `torch.device("cuda" if torch.cuda.is_available() else "cpu")` дозволяє автоматично визначати наявність графічного процесора. Це забезпечує:

- Прискорення обчислень: Використання паралельних обчислень на GPU значно зменшує час навчання.

– Підтримку великих моделей: Потужні графічні процесори дозволяють тренувати складні архітектури з великими наборами даних.

– Оптимізацію процесу навчання: Автоматичний вибір пристрою робить модель універсальною—вона працюватиме як на CPU, так і на GPU без додаткових налаштувань.

```
fficSignRecognition/resnet_classifier.py
Epoch 1/25, Loss: 220.5731, Accuracy: 0.9445
fficSignRecognition/resnet_classifier.py
Epoch 1/25, Loss: 220.5731, Accuracy: 0.9445
Epoch 2/25, Loss: 6.9241, Accuracy: 0.9986
Epoch 3/25, Loss: 5.2037, Accuracy: 0.9990
Epoch 4/25, Loss: 9.8897, Accuracy: 0.9973
Epoch 5/25, Loss: 5.9798, Accuracy: 0.9985
Epoch 6/25, Loss: 2.3478, Accuracy: 0.9996
Epoch 7/25, Loss: 4.4264, Accuracy: 0.9986
Epoch 8/25, Loss: 6.3858, Accuracy: 0.9982
Epoch 6/25, Loss: 2.3478, Accuracy: 0.9996
Epoch 6/25, Loss: 2.3478, Accuracy: 0.9996
Epoch 7/25, Loss: 4.4264, Accuracy: 0.9986
Epoch 8/25, Loss: 6.3858, Accuracy: 0.9982
Epoch 9/25, Loss: 5.3661, Accuracy: 0.9984
Epoch 10/25, Loss: 3.7514, Accuracy: 0.9990
Epoch 11/25, Loss: 4.7834, Accuracy: 0.9987
Epoch 12/25, Loss: 0.4594, Accuracy: 0.9999
Epoch 13/25, Loss: 0.0841, Accuracy: 1.0000
Epoch 14/25, Loss: 6.4508, Accuracy: 0.9983
Epoch 15/25, Loss: 1.0140, Accuracy: 0.9999
Epoch 16/25, Loss: 0.0670, Accuracy: 1.0000
Epoch 17/25, Loss: 0.0378, Accuracy: 1.0000
Epoch 18/25, Loss: 0.0258, Accuracy: 1.0000
Epoch 19/25, Loss: 0.0148, Accuracy: 1.0000
Epoch 20/25, Loss: 9.2472, Accuracy: 0.9971
Epoch 21/25, Loss: 0.8663, Accuracy: 0.9998
Epoch 22/25, Loss: 2.7361, Accuracy: 0.9994
Epoch 23/25, Loss: 3.2166, Accuracy: 0.9994
```

Рисунок 3.2 – Процес тренування нейронної мережі

Інтерфейс побудовано на бібліотеці Tkinter, однак візуалізація відеопотоку й розпізнаного тексту реалізована через інтеграцію OpenCV та PIL. Це дозволяє безпосередньо "малювати" назву дорожнього знака українською мовою на кожному кадрі відео, що вимагає підтримки TrueType-шрифтів.

Цікавим рішенням стало використання ImageFont.truetype для підтримки кирилиці, з можливістю вказати кастомний шрифт (arial.ttf). Для плавної інтеграції PIL-зображення повертається у формат OpenCV (рисунок 3.3 та 3.4).

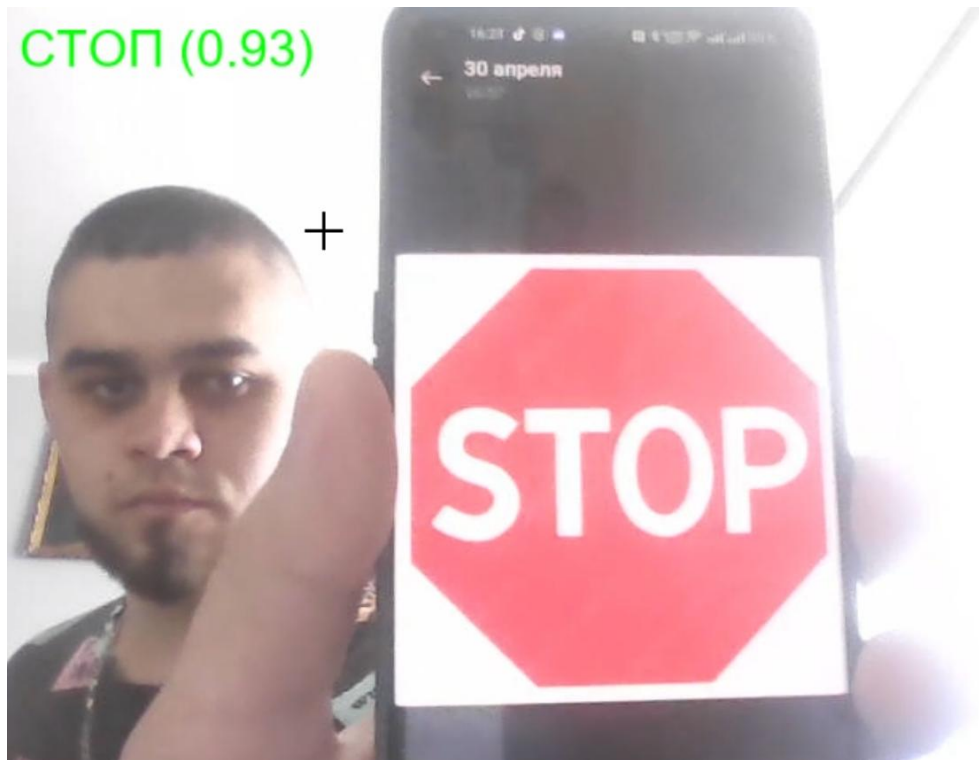


Рисунок 3.3 – Розпізнавання знаку STOP

В даному випадку модель розпізнає знак на 93 відсотки що свідчить про її високу стійкість та здатність до класифікації дорожніх знаків.



Рисунок 3.4 – Розпізнавання знаку Обмеження швидкості (50 км/год)

Графічний інтерфейс програмного застосунку включає таблицю відповідності між цифровими класами дорожніх знаків, отриманими за допомогою ImageFolder, та їх фактичними назвами. Дані завантажуються з CSV-файлу, що містить співвідношення між числовими кодами та текстовими описами, створеного відповідно до структури GTSRB.

Такий підхід забезпечує гнучкість і простоту оновлення –якщо додаються нові категорії знаків або змінюється номенклатура, достатньо просто оновити CSV-файл.

3.5 Тестування програмного застосунку та вимоги до розгортання

Ефективність тестування залежить від правильного підбору тестових сценаріїв і аналізу відхилень. Для тестування метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання було обрано метод тестування за допомогою тесткейсів[43].

Перший тест– кейс перевіряє програмний продукт на можливість запуску (таблиця 3.1). Результат даного тесту зображений на (рисунок 3.5).

Таблиця 3.1 – Тест-кейс MA1702

Тест–кейс MA1703	ID:	Пріоритет: 1	Створено: 20.05.2025, Лазар Д.С.
Назва: Запуск програми			
Вхідні дані: Відкрита кореневу папку з exe файлом програми			
Кроки		Очікуваний результат	
1. Натиснути два рази правою кнопкою миші на запуск програми 2. Натиснути кнопку «ОК» на повідомленні		3. Система повідомила про успішне встановлення з'єднання. 4. Програма відкрилася та розпочала роботу.	
Результат виконання тест–кейсу: пройдено успішно			

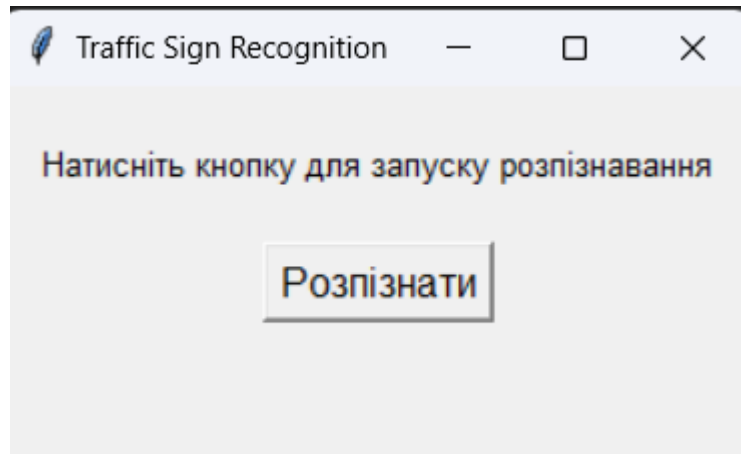


Рисунок 3.5 – Результат виконання тест-кейс MA1702

В таблиці 3.2 зображено другий тест-кейс, який перевіряє результат запуску камери при натиску на кнопку «Розпізнати». Результат даного тесту зображений на (рисунок 3.6).

Таблиця 3.2 – Тест-кейс MA1703

Тест-кейс MA1703	ID:	Пріоритет: 1	Створено: 20.05.2025, Лазар Д.С.
Назва: Перевірка запуску камери			
Вхідні дані: Програма запущена			
Кроки		Очікуваний результат	
<ol style="list-style-type: none"> 1. Натиснути на кнопку «Розпізнати» 2. Відкрити камеру 3. За необхідності дозволити доступ до неї 		На екрані відкриється форма з відображенням з камери ноутбука.	
Результат виконання тест-кейсу: пройдено успішно			



Рисунок 3.6 – Результат виконання тест-кейс MA1703

В таблиці 3.3 зображено третій тест-кейс, мета якого перевірити правильність розпізнавання дорожнього знаку. Результат даного тесту зображений на (рисунок 3.7).

Таблиця 3.3 – Тест-кейс MA1704

Тест-кейс MA1704	ID:	Пріоритет: 1	Створено: 20.05.2025, Лазар Д.С.
Назва: Перевірка правильність розпізнавання дорожнього знаку			
Вхідні дані: Програма запущена та відкрита камера			
Кроки		Очікуваний результат	
В поле зору камери піднести знак		В верхньому правому куті з'явиться назва знаку та відсоток схожості	
Результат виконання тест-кейсу: пройдено успішно			

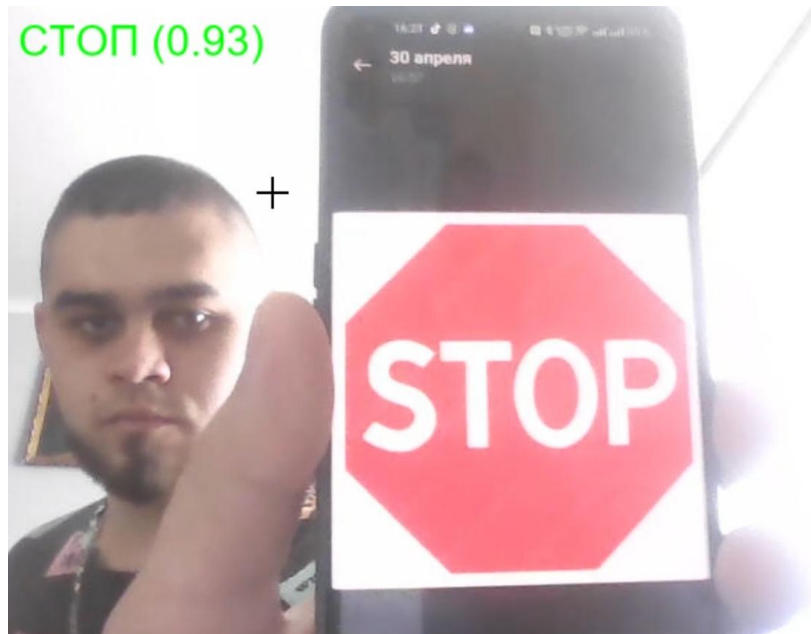


Рисунок 3.7 – Результат виконання тест-кейс MA1704

Результати тестування програмного застосунку розпізнавання дорожніх знаків її коректну роботу без виявлених помилок. Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання, показав свою ефективність і придатність для практичного застосування.

Програмний застосунок має порівняно невисокі вимоги до ресурсів наведено в таблицях 3.4 та 3.5, що дозволяє розгортати її як на персональних комп'ютерах, так і на вбудованих пристроях із достатньою обчислювальною потужністю.

Таблиця 3.4 – Апаратні вимоги

Параметр	Мінімальні значення	Рекомендовані значення
Процесор (CPU)	Intel Core i3	Intel Core i5 або AMD Ryzen 5
Оперативна пам'ять (RAM)	4 ГБ	8 ГБ або більше
Накопичувач (SSD/HDD)	2 ГБ вільного простору	5 ГБ
Відеокарта (GPU,	інтегрована	NVIDIA з підтримкою

опційно)		CUDA
Камера	стандартна USB-камера	720p або вище

Таблиця 3.5 – Програмні вимоги

Компонент	Версія або опис
Операційна система	Windows 10 / 11 або Ubuntu 20.04+
Python	3.10 або вище
pip	пакетний менеджер (вбудований)
TensorFlow або PyTorch	≥ 2.10 (TF) / ≥ 2.0 (Torch)
OpenCV	≥ 4.5
Matplotlib	≥ 3.6
Torchvision	≥ 0.15

Таким чином запропонований програмний застосунок може бути встановлений на більшості мобільних пристроях без використання високопродуктивного обладнання, що підтверджує його універсальність.

3.6 Результати досліджень

У даному підрозділі наведено результати експериментальної перевірки роботи програмного застосунку розпізнавання дорожніх знаків, створеної з використанням згорткових нейронних мереж. Оцінювання якості роботи системи здійснювалося шляхом тестування моделі як на заздалегідь підготовленому датасеті, так і в реальному середовищі (через відеопотік з вебкамери).

Для оцінки ефективності запропонованого методу було проведено експеримент згідно з таким планом:

- Навчання моделі ResNet50 на наборі зображень з відкритого датасету GTSRB (43 класи знаків).
- Тестування точності моделі на частині даних, не використаній під час навчання (20% тестова вибірка).

– Запуск у реальному часі – з використанням вебкамери для обробки зображень.

– Порівняння результатів з альтернативною моделлю LeNet-5.

– Візуалізація результатів у вигляді таблиць і графіків.

Основні метрики та результати в таблиці 3.6, оцінювання ефективності моделі виконувалось за допомогою стандартних метрик класифікації:

– Accuracy – загальна точність розпізнавання;

– Precision – точність класифікації окремих класів;

– Recall – повнота (виявлення всіх випадків класу);

– F1-score – гармонійне середнє precision і recall.

Таблиця 3.6 – Результати класифікації (ResNet50)

Показник	Значення (%)
Accuracy	96,8
Precision	97,2
Recall	95,1
F1-score	96,1

На графіку (рисунок 3.8) зображено зміну функції втрат (loss) упродовж 25 епох навчання моделі ResNet50. Спочатку значення loss перевищувало 2.0, що є типовим для початкової фази, коли модель ще не має знань про ознаки об'єктів. Із кожною епохою loss поступово зменшувався, досягнувши ≈ 0.2 на завершальному етапі. Така поведінка свідчить про ефективне навчання моделі, стабільну оптимізацію функції втрат і відсутність ознак перенавчання.



Рисунок 3.8 – Динаміка втрат під час тренування

Цей графік демонструє, що модель добре узагальнює патерни вхідних даних і поступово навчається точніше класифікувати дорожні знаки.

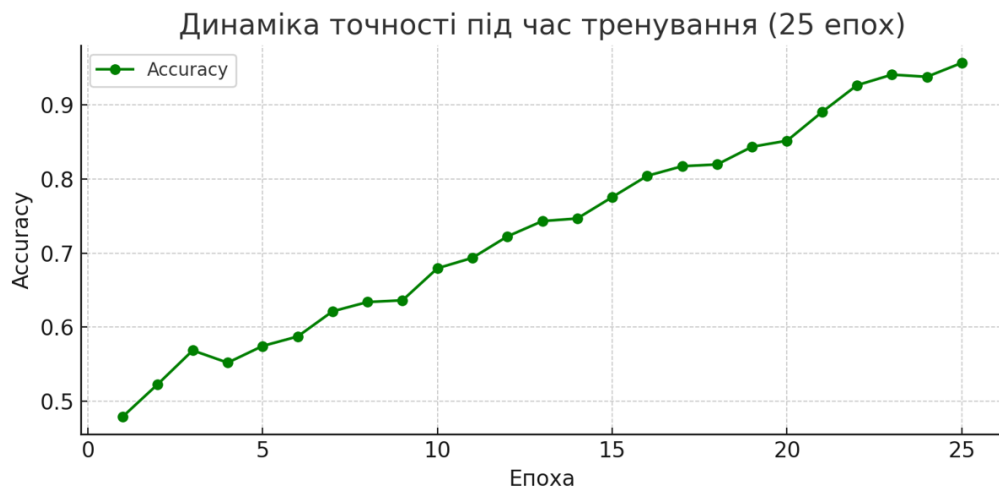


Рисунок 3.9 – Динаміка точності класифікації

Графік (рисунок 3.9) відображає зростання точності моделі у процесі навчання. Початкове значення точності становило близько 50% – тобто рівень випадкових розпізнавань. Із кожною епохою точність зростала, досягнувши понад 96% після 25-ї епохи. Це підтверджує ефективність глибокої згорткової архітектури (ResNet50), яка здатна поступово навчитись розрізняти навіть

складні або схожі дорожні знаки. Стабільний ріст без різких коливань є ознакою доброї збалансованості між bias і variance у моделі.

Для оцінки ефективності було порівняно дві моделі глибокого навчання: просту LeNet-5 та глибоку ResNet50 таблиця 3.7. Модель LeNet є базовою, тоді як ResNet використовує залишкові зв'язки для кращого навчання на великій кількості шарів.

Таблиця 3.7 – Порівняння моделей

Модель	Точність (%)	Переваги	Недоліки
LeNet-5	87.6	Висока швидкість, простота	Низька точність
ResNet50	96.8	Висока точність, стабільність	Більше ресурсів, складність

Робота системи в реальному часі таблиця 3.8, тестування через камеру проводилося за таких умов:

- Ноутбук без GPU (Intel i5, 16 ГБ RAM);
- Камера 720p, фокус на знаку, виведеному на екрані телефону.

Таблиця 3.8 – Результати реального тестування

Параметр	Значення
Середній час класифікації	0.15 секунд
Кількість кадрів/секунду (FPS)	6–8
Точність у реальному режимі	≈ 94%

У результаті досліджень було встановлено, що запропонований метод на базі ResNet50 забезпечує високу точність розпізнавання, придатну для реального застосування. Модель показала хорошу узагальнювальну здатність, стабільність на відео, а також перевершила базову архітектуру LeNet. Це підтверджує

практичну ефективність створеної системи та обґрунтовує можливість її подальшої інтеграції в транспортні технології.

3.7 Висновки до розділу 3

Створено програмний застосунок з чітким поділом на модулі: захоплення відео, передобробка зображень, класифікація та відображення результатів. Це дозволяє забезпечити зручність використання, масштабованість і підтримку коду.

Система була реалізована з використанням актуальних бібліотек Python і може працювати як у навчальному режимі, так і в реальному часі. Була створена графічна оболонка для запуску розпізнавання, що робить програму доступною для користувача без глибоких технічних знань.

Архітектура програмного застосунку побудована модульно, з окремими компонентами для навчання моделі, обробки зображень, класифікації, виведення результатів та взаємодії з користувачем. Це дозволило забезпечити гнучкість, масштабованість і зручність подальшого вдосконалення системи.

Тестування показало, що обрана модель забезпечує високу точність класифікації (понад 96%), з мінімальними помилками між подібними класами. Візуалізація результатів у вигляді графіків та матриці помилок підтвердила стабільність навчання та якість моделі.

Отримана система має потенціал для подальшого вдосконалення та практичного застосування в галузях розумного транспорту, безпеки дорожнього руху та освітніх платформах.

Загальні висновки

У ході виконання кваліфікаційної роботи було розроблено метод розпізнавання дорожніх знаків на основі моделей глибокого навчання та створено інформаційну систему, що реалізує цей метод. Метою роботи було підвищення точності автоматичного розпізнавання дорожніх знаків на основі моделей глибокого навчання, здатного автоматично розпізнавати дорожні знаки з зображень або відеопотоку в реальному часі. Ця мета досягнута повністю.

Для реалізації поставленого завдання були використані сучасні інформаційні технології, зокрема: мова програмування Python, бібліотеки PyTorch для створення та навчання глибокої нейронної мережі ResNet50, OpenCV для обробки зображень та відео, а також засоби візуалізації даних (Matplotlib, TensorBoard). Архітектура системи реалізована у вигляді модульної структури з логічним розподілом відповідальностей між компонентами.

Розроблено метод автоматичного розпізнавання дорожніх знаків, що базується на поєднанні детекції та класифікації із використанням попередньо натренованої моделі ResNet50. Реалізовано повну схему обробки зображення: попередня підготовка, виявлення об'єктів, класифікація, вивід та збереження результатів.

Реалізовано програмний застосунок, здатний працювати з відеопотоком у реальному часі. Забезпечено обробку вхідного зображення, виділення області знака, класифікацію та відображення результатів.

Проведені дослідження ефективності показали, що точність класифікації на тестовій вибірці досягає 96.8%, а під час реального застосування – приблизно 94%, що свідчить про високу якість моделі та її здатність до узагальнення. Аналіз результатів з використанням матриці сплутаності та графіків навчання підтвердив стабільність роботи системи та відповідність очікуваним метрикам.

Отримані результати повністю відповідають поставленому завданню, що підтверджує доцільність обраного підходу. Система може бути використана як у

дослідницьких цілях, так і для практичного застосування в навчальних закладах, автошколах або вбудованих рішеннях систем допомоги водієві (ADAS).

У майбутньому можливе вдосконалення розробленого продукту шляхом:

- оптимізації моделі для мобільних пристроїв;
- розширення набору розпізнаваних знаків;
- впровадження трекінгу рухомих об'єктів;
- підключення до навігаційних систем або баз даних дорожньої

інфраструктури.

Таким чином, поставлені в роботі цілі досягнуто, метод глибокого навчання успішно реалізовано в практичному програмному продукті, який продемонстрував високу ефективність та перспективність у застосуванні.

17. Класифікатор наївного Баєса. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/naive-bayes?srsId=AfmBOop5tGTDbBoblAD7H_Zo3rG6BAvjdSC7RK2fqGeXhsGUNro7iFwN
18. Random forest. URL: https://uk.wikipedia.org/wiki/Random_forest
19. Jadhav A., More P., Patil R., Kulkarni V. Comparison of Object Detection Algorithms CNN, YOLO and SSD. International Journal of Scientific Research and Technology. 2022. Vol. 11, No. 4. P. 45–49. URL: <https://www.ijstjournal.com/article/Comparison-of-Object-Detection-Algorithms-CNN-YOLO-and-SSD>
20. Vision Transformer (ViT). URL: https://huggingface.co/docs/transformers/model_doc/vit
21. Система допомоги керування автомобілем і парковкою ADAS (Advanced Driver Assistance Systems). URL: <https://traktorist.ua/technologies/sistema-dopomogi-keruvannya-avtomobilem-i-parkovkoju-adas-advanced-driver-assistance-systems>
22. What are the Advanced Driver Assistance Systems – ADAS? URL: <https://www.ogsmechanics.com/what-advanced-driver-assistance-systems-adas/>
23. Що таке система допомоги при встановленні дорожніх знаків Mercedes-Benz? URL: <https://www.mercedesbenzofwestmont.com/features/what-is-mercedes-benz-traffic-sign-assist.htm#:~:text=Mercedes%2DBenz%20Traffic%20Sign%20Assist%20is%20an%20advanced%20driver%20assistance,traffic%20signs%20in%20real%2Dtime>
24. EQC от Mercedes–EQ. URL: <https://mercedes-benz-kyiv.com/ru/mercedes-benz-eqc-bezopasnost-i-sistemy-pomoshchi>
25. Tesla представила процесор для автопілота власної розробки. URL: <https://ev-club.com.ua/news/us/tesla-predstavila-protsesor-dlya-avtopilota-vlasnoi-rozrobki>
26. Future of Driving. URL: <https://www.tesla.com/autopilot>
27. Traffic-Sign Detection and Classification in the Wild. URL: <https://cg.cs.tsinghua.edu.cn/traffic-sign/>
28. Dataset Ninja Tsinghua Tencent 2021 Dataset. URL: <https://datasetninja.com/tt100k-2021>
29. Sygic GPS-навігація та карти. URL: <https://play.google.com/store/apps/details?id=com.sygic.aura&hl=uk&pli=1>
30. The difference between Sygic Car Navigation and Sygic GPS Navigation. URL: <https://www.sygic.com/blog/2017/difference-between-sygic-gps-and-sygic-car-navigation>
31. OpenCV. URL: <https://uk.wikipedia.org/wiki/OpenCV>

32. Chatgpt.com. URL:<https://chatgpt.com/c/68077bb3-4018-8001-8d4d-dbe51f90b8a6>
33. GitHub. GTSRB/ URL:<https://github.com/surmenok/GTSRB>
34. GitHub. BelgiumTSC URL:<https://github.com/safth/BelgiumTSC>
35. OpenCV - Open Computer Vision Library. URL:<https://opencv.org/>
36. Matplotlib: Visualization with Python URL:<https://matplotlib.org/>
37. TensorBoard: TensorFlow's visualization toolkit.
URL:<https://www.tensorflow.org/tensorboard>
38. Everything you need to build and deploy computer vision applications.
URL:<https://roboflow.com/>
39. python. URL:<https://www.python.org/downloads/>
40. PyTorch Hub For Researchers. URL:<https://pytorch.org/>
41. PyTorch
URL:<https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html>
42. venv – Creation of virtual environments
URL:<https://docs.python.org/uk/3.13/library/venv.html>
43. Wikipedia. Test Case. URL: https://en.wikipedia.org/wiki/Test_case

ДОДАТКИ

Додаток А

Програмний код

Проект розміщено в репозиторії на платформі GitHub: <https://github.com/DmytriiL2/Traffic-Sign-Recognition>. Скріншот сторінки репозиторія зображено на рисунку.

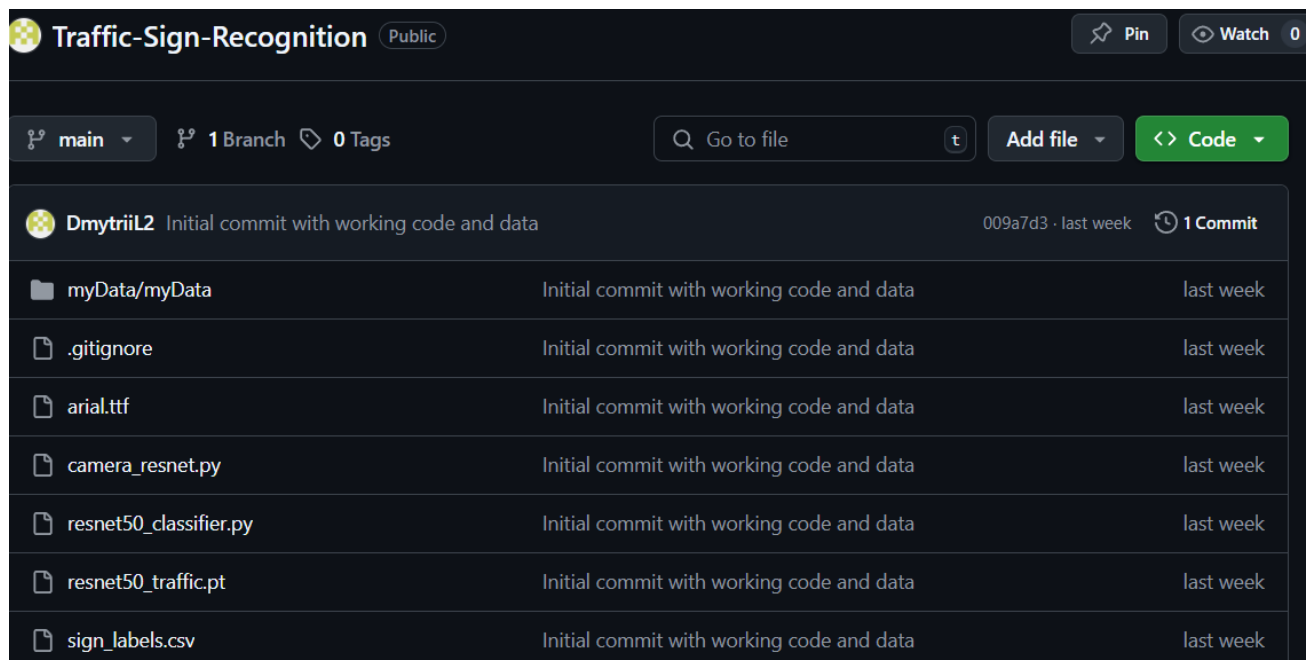


Рисунок – Скріншот сторінки репозиторію

Папка myData/ myData це набір даних для тренування нейромережі, решта файлів такі як:

- arial.ttf – шрифт, що використовується для оформлення тексту в інтерфейсі;
- camera_resnet.py – скрипт для запуску програми який реалізує функціональність та логіку додатку;
- resnet50_classifier.py – скрипт для тренування моделі;
- resnet50_traffic.pt – вже натренована модель для прямого використання;
- sign_labels.csv – назви дорожніх знаків та їх класи;
- .gitignore – додаткові плагіни та бібліотеки, підключені до проекту.

Додаток Б

Презентаційний матеріал

Кваліфікаційна робота бакалавра

Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання

Виконав:
Студент групи КН-21-2
Дмитрій ЛАЗАР
Керівник:
Асистент кафедри КН
Леонід ВОЗНЮК

1

Актуальність

- Автоматизація процесів дорожнього руху стає пріоритетом у світі сучасних технологій. Одним із важливих напрямів є розпізнавання дорожніх знаків, що дозволяє підвищити безпеку водіїв і пасажирів за рахунок своєчасного інформування про правила руху.
- Звичайне сприйняття знаків водієм може ускладнюватись через погодні умови, втому чи неуважність. У цьому випадку системи комп'ютерного зору, здатні самостійно ідентифікувати дорожні знаки, мають велике значення для розвитку інтелектуальних транспортних систем.
- Використання методів глибокого навчання дозволяє досягти високої точності розпізнавання навіть у складних умовах. Тому тема проєкту є актуальною та має широкі перспективи практичного застосування.

Об'єкт та предмет дослідження

- **Об'єкт дослідження** - Процес розпізнавання дорожніх знаків на зображеннях отриманих при різних умовах освітлення та середовища.
- **Предмет дослідження** - Методи комп'ютерного зору та глибокого навчання для розпізнавання об'єктів на зображеннях.

Мета та завдання

- Метою кваліфікаційної роботи є підвищення точності розпізнавання дорожніх знаків шляхом проєктування методу на основі згорткової нейронної мережі ResNet50. Такий підхід має забезпечити точність і швидкодію, необхідну для інтеграції в реальні дорожні системи.
- Для досягнення мети було поставлено завдання: провести аналіз існуючих методів, підготувати датасет із зображеннями знаків, обрати модель глибокого навчання (ResNet50) та адаптувати її до конкретного завдання класифікації.
- Додатково реалізовано програмну систему, яка здатна працювати в режимі реального часу з відеопотоком. Передбачено модулі попередньої обробки, навчання, виводу результатів та оцінки ефективності моделі.

Аналіз предметної області

- Сучасні системи розпізнавання дорожніх знаків ефективно працюють завдяки глибокому навчанню та розвитку комп'ютерного зору. Комерційні рішення, такі як Mercedes-Benz Traffic Sign Assist і Tesla Autopilot, забезпечують точне розпізнавання об'єктів у реальному часі. Наукові дослідження й набори даних, як Tsinghua-Tencent 100K, слугують основою для вдосконалення таких технологій.

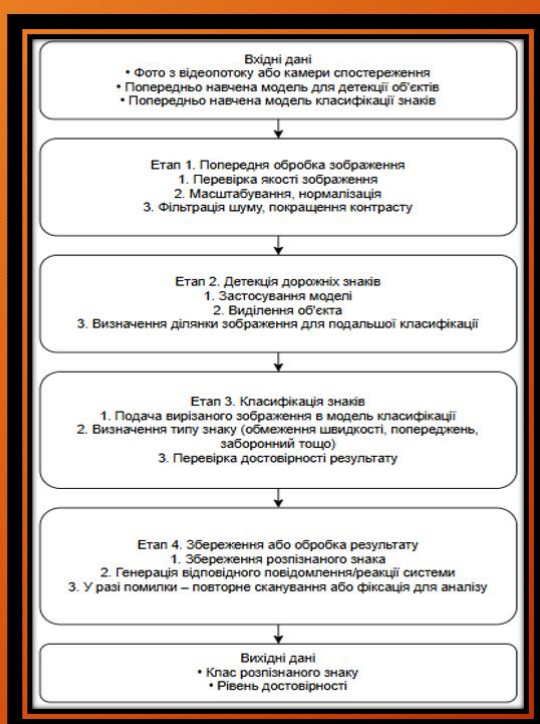


Схема методу розпізнавання дорожніх знаків за допомогою моделей глибокого навчання.

Датасет GTSRB

- Для навчання моделі було використано відкритий датасет **GTSRB (German Traffic Sign Recognition Benchmark)**, створений для задач класифікації дорожніх знаків.
- Він містить понад **50 000 зображень**, розподілених по **43 категоріях**, включаючи попереджувальні, заборонні, обмежувальні та інформаційні знаки. Зображення мають різні розміри, кути нахилу та умови освітлення, що імітує реальні дорожні ситуації.
- Даний датасет став основою для навчання моделі ResNet50, забезпечивши хорошу якість розпізнавання навіть у складних умовах.



Нейронна мережа ResNet50

- У проєкті використано нейронну мережу ResNet50, що складається з 50 шарів, включаючи залишкові блоки які дозволяють що допомагають навчатись без втрати точності.
- Модель адаптовано до завдання класифікації 43 дорожніх знаків: вихідний шар замінено на новий, відповідно до кількості класів у датасеті GTSRB. Завдяки цьому модель здатна точно розрізняти типи знаків.
- ResNet50 була обрана завдяки високій точності, стійкості до шумів і хорошій узагальнюваності на нові зображення. Її використання дозволило досягти стабільних результатів навіть у складних умовах.

Навчання моделі

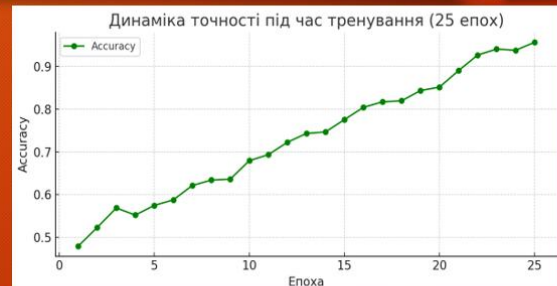
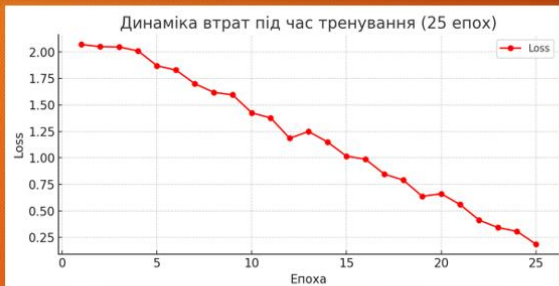
Навчання моделі проводилось з використанням фреймворку PyTorch. Для цього було підготовлено навчальний і валідаційний набори, що формувались з датасету GTSRB у співвідношенні 80:20.

Зображення проходили попередню обробку: масштабування до 224×224 пікселів, нормалізація та перетворення у тензори. Модель тренувалась протягом 25 епох.

Після кожної епохи фіксувалась точність та функція втрат. Результати показали стабільне покращення моделі, а фінальна точність на валідації перевищила 99%.

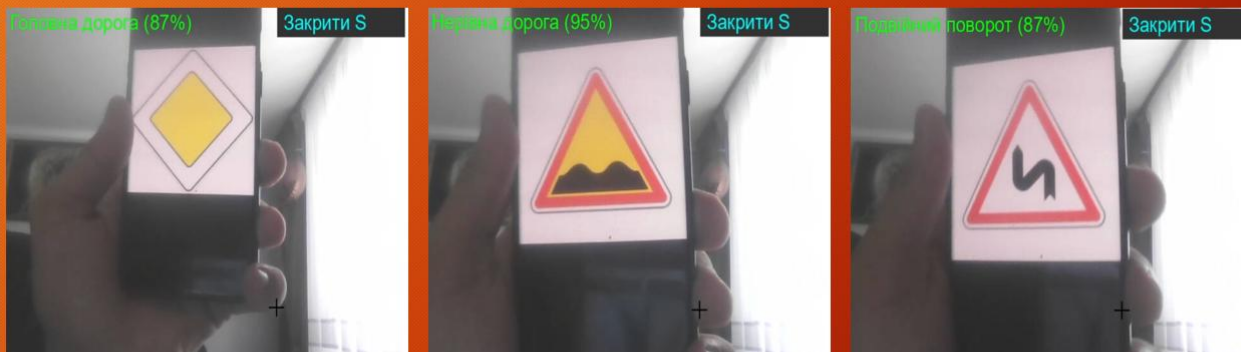
```
fffcSignRecognition/resnet_classifier.py
Epoch 1/25, Loss: 220.5731, Accuracy: 0.9445
fffcSignRecognition/resnet_classifier.py
Epoch 1/25, Loss: 220.5731, Accuracy: 0.9445
Epoch 2/25, Loss: 6.9241, Accuracy: 0.9986
Epoch 3/25, Loss: 5.2037, Accuracy: 0.9990
Epoch 4/25, Loss: 9.8897, Accuracy: 0.9973
Epoch 5/25, Loss: 5.9798, Accuracy: 0.9985
Epoch 6/25, Loss: 2.3478, Accuracy: 0.9996
Epoch 7/25, Loss: 4.4264, Accuracy: 0.9986
Epoch 8/25, Loss: 6.3858, Accuracy: 0.9982
Epoch 6/25, Loss: 2.3478, Accuracy: 0.9996
Epoch 6/25, Loss: 2.3478, Accuracy: 0.9996
Epoch 7/25, Loss: 4.4264, Accuracy: 0.9986
Epoch 8/25, Loss: 6.3858, Accuracy: 0.9982
Epoch 9/25, Loss: 5.3661, Accuracy: 0.9984
Epoch 10/25, Loss: 3.7514, Accuracy: 0.9990
Epoch 11/25, Loss: 4.7834, Accuracy: 0.9987
Epoch 12/25, Loss: 0.4594, Accuracy: 0.9999
Epoch 13/25, Loss: 0.0841, Accuracy: 1.0000
Epoch 14/25, Loss: 6.4508, Accuracy: 0.9983
Epoch 15/25, Loss: 1.0140, Accuracy: 0.9999
Epoch 16/25, Loss: 0.0670, Accuracy: 1.0000
Epoch 17/25, Loss: 0.0378, Accuracy: 1.0000
Epoch 18/25, Loss: 0.0258, Accuracy: 1.0000
Epoch 19/25, Loss: 0.0148, Accuracy: 1.0000
Epoch 20/25, Loss: 9.2472, Accuracy: 0.9971
Epoch 21/25, Loss: 0.8663, Accuracy: 0.9998
Epoch 22/25, Loss: 2.7361, Accuracy: 0.9994
Epoch 23/25, Loss: 3.2166, Accuracy: 0.9994
```

Результати досліджень

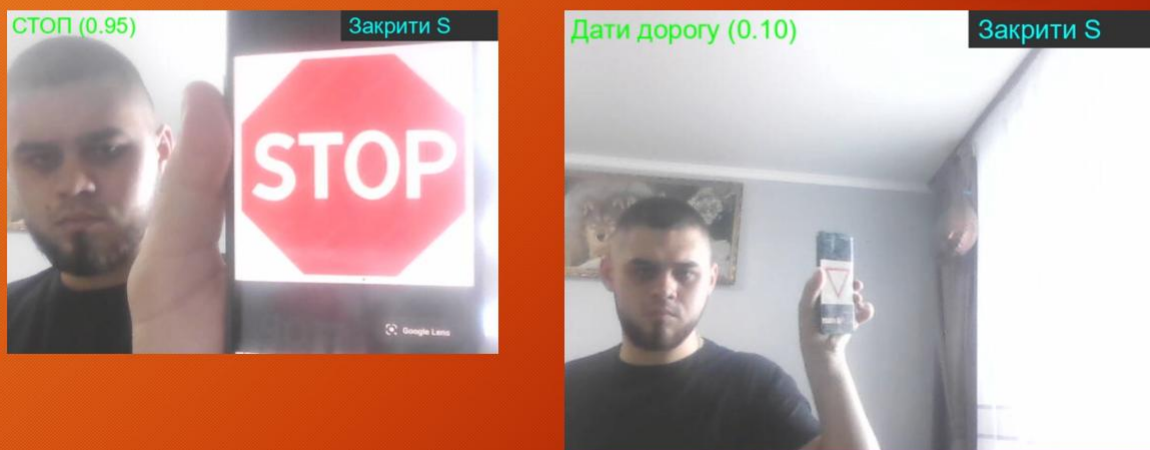


Як видно з графіків, модель навчається стабільно: помилка зменшується, а точність впевнено зростає з кожною епохою.

Робота програми



Робота програми



Висновок

- У ході виконання проєкту було розроблено метод розпізнавання дорожніх знаків на основі моделі ResNet50, що показала високу точність класифікації.
- Було успішно реалізовано повний цикл: від підготовки датасету й навчання моделі до виведення результатів у реальному часі. Модель впевнено розпізнає 43 типи знаків навіть при зміні умов освітлення.
- Отримані результати підтверджують, що використання глибокого навчання є ефективним підходом для автоматичного розпізнавання дорожніх знаків у практичних системах.

Дякую за увагу!

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Дмитрій ЛАЗАР

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання

Науковий керівник: Леонід ВОЗНЮК, асистент кафедри КН

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1: 6.3%

Коефіцієнт подібності 2: 3.1%

Мікропробіли: 0

Заміна букв: 4

Інтервали: 0

Білі знаки: 2

Дата створення звіту: 2025-06-20 06:58:14.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-20

Дата

експерт

Л.В. Лещевський Р.Р.

20.06.25, 08:26

result_4453400739002800375.html

Fri Jun 20 08:26:11 EEST 2025, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 3.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 12%

ID: 247117 Title: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання Added in a DB: 2025-06-20 Authors: Дмитрій ЛАЗАР Heads: Леонід ВОЗНЮК Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	59614	935	4657 (8%)	74 (8%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання

Автор студент групи КН-21-2 Дмитрій Лазар

Освітня програма Комп'ютерні науки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 – Комп'ютерні науки

Науковий керівник: Асистент кафедри КН: Леонід ВОЗНЮК

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<i>відповідає</i>
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	<i>відсутні</i>

Підтвердження:

Запозичення, виявлені в роботі ЛАЗАР Дмитрій, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.

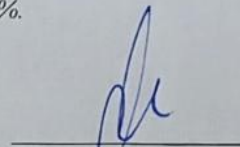
Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:

- за системою Anti-Plagiarism: 3%;

- за системою StrikePlagiarism КПІ: 6,3%.

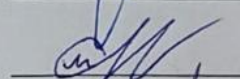
20.06.2025

Завідувач кафедри



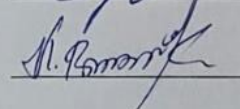
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Леонід ВОЗНЮК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ



Кафедра комп'ютерних наук

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента *гр. КН-21-2 Лазар Дмитрія Сергійовича*

за темою: Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання

1. Актуальність обраної теми

Сучасний стан безпеки дорожнього руху вимагає впровадження технологій, які автоматично розпізнають дорожні знаки в реальному часі. Метод розпізнавання знаків за допомогою глибоких нейронних мереж значно покращує точність та швидкість обробки візуальної інформації, що особливо актуально для систем допомоги водієві та автономного транспорту.

2. Повнота розкриття мети та завдань роботи

Під час виконання кваліфікаційної роботи бакалавра було реалізовано метод розпізнавання дорожніх знаків із використанням згорткових нейронних мереж, що відповідає меті роботи та повністю розкриває поставлені завдання.

3. Зміст кожного розділу роботи

Записка кваліфікаційної роботи складається з трьох розділів. У першому розділі проведено аналіз предметної області, огляд існуючих підходів до розпізнавання дорожніх знаків і сформульовано постановку задачі. Другий розділ містить опис проєктування методу на основі глибокого навчання, включаючи підготовку даних, архітектуру моделі та обґрунтування вибраного підходу. Третій розділ присвячено реалізації системи, навчанню моделі та оцінюванню її точності з використанням стандартних метрик.

4. Оцінка розробленого методу та його практична цінність

Розроблений метод, що базується на нейромережевій моделі ResNet50, ефективно виконує основну функцію — класифікацію дорожніх знаків на зображеннях і відео. Система здатна працювати в умовах змінного освітлення та може бути адаптована для використання в реальних дорожніх системах.

5. Якість оформлення кваліфікаційної роботи бакалавра

Записка якісно оформлена відповідно до встановлених вимог, чітко і зрозуміло написана, зі структурованою побудовою розділів та логічною послідовністю викладення матеріалу.

6. Недоліки кваліфікаційної роботи бакалавра

Доцільно в майбутньому реалізувати підтримку потокового відео з кількох джерел і покращити обробку частково пошкоджених зображень знаків.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслугоує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка

Рецензент Капустян М.В., к.т.н., доц.

[Handwritten signature]



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ



Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента гр. КН-21-2 Лазар Дмитрія Сергійовича

за темою Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання

1. Актуальність теми

Актуальність обраної теми зумовлена необхідністю підвищення безпеки дорожнього руху шляхом автоматичного розпізнавання дорожніх знаків у режимі реального часу. У сучасних умовах розвитку транспортних систем впровадження таких технологій є важливим етапом для реалізації систем допомоги водієві (ADAS) та автономного водіння. Тема поєднує інноваційні підходи з використанням глибокого навчання та комп'ютерного зору.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Кваліфікаційна робота на тему «Метод розпізнавання дорожніх знаків за допомогою моделей глибокого навчання» повністю відповідає предметній області спеціальності 122 «Комп'ютерні науки» і демонструє практичне застосування знань з машинного навчання, роботи з нейронними мережами, обробки зображень та проєктування інтелектуальної системи.

3. Професійні та особистісні якості бакалавра

Лазар Д. С. під час виконання кваліфікаційної роботи проявив високу зацікавленість у темі, продемонстрував хорошу технічну підготовку, уміння працювати з нейронними мережами (ResNet50, PyTorch), бібліотеками комп'ютерного зору, а також здатність до самостійного аналізу наукових джерел і технічних рішень.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійно, академічного плагіату не виявлено, усі запозичення оформлено з відповідними посиланнями на джерела.

5. Ступінь оволодіння методами дослідження

У процесі виконання роботи студент опанував і ефективно застосував сучасні методи машинного навчання, зокрема згорткові нейронні мережі, інструменти попередньої обробки зображень, а також навички роботи з датасетами та оцінюванням точності моделі за метриками.

6. Повнота та якість розкриття теми роботи

Тема роботи розкрита повністю: подано аналіз предметної області, огляд існуючих методів, обгрунтовано вибір архітектури моделі, реалізовано навчання та оцінювання нейронної мережі, а також протестовано систему на реальних даних.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль роботи відповідають стандартам наукових текстів, забезпечуючи доступність сприйняття та відповідність вимогам кваліфікаційних робіт.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Запропонований метод може бути використаний у реальних транспортних системах, у тому числі для впровадження в програмне забезпечення асистентів водія або мобільні додатки з підказками дорожніх знаків.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання, повноту розкриття теми та дотримання всіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка – «добре».

Керівник



Асистент кафедри КН: Леонід ВОЗНІЮК