

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Система керування транспортним рухом з використанням технології
комп'ютерного зору та індукційної петлі»

на здобуття освітнього ступеня бакалавра

зі спеціальності 126 Інформаційні системи та технології

(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології

(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Віктор ШЕВЧУК

Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. ІСД-43 Віктор ШЕВЧУК
(Ім'я, прізвище)

Керівник: PhD Валентина ДАНИЛЬЧЕНКО
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент:
(науковий ступінь, вчене звання)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

_____ Ім'я, ПРИЗВИЩЕ

«_____» _____ 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Шевчук Віктор Володимирович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Система керування транспортним рухом з використанням технології комп'ютерного зору та індукційної петлі

керівник кваліфікаційної роботи PhD Валентина ДАНИЛЬЧЕНКО,

(Ім'я, ПРИЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56

2. Строк подання кваліфікаційної роботи «30» травня 2025р.

3. Вихідні дані до кваліфікаційної роботи: Науково технічна література з теми бакалаврської роботи, Аналіз існуючих систем для керування дорожнім рухом.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз сучасного стану систем керування транспортним рухом
2. Програмна реалізація системи керування транспортним рухом
3. Моделювання та тестування програмного рішення

5. Перелік ілюстративного матеріалу: презентація

6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Опрацювання теми, постановка мети та завдань, підготовка вступу	24.02.2025 – 02.03.2025	
2	Написання розділу 1: Аналіз сучасного стану систем керування транспортним рухом	03.03.2025 – 12.03.2025	
3	Написання розділу 2: Програмна реалізація системи керування транспортним рухом	13.03.2025 – 24.03.2025	
4	Написання розділу 3: Моделювання та тестування програмного рішення	25.03.2025 – 07.04.2025	
5	Проведення аналізу результатів, підготовка висновків	08.04.2025 – 12.04.2025	
6	Підготовка графічного матеріалу, ілюстрацій, рисунків	13.04.2025 – 17.04.2025	
7	Редагування, перевірка змісту, перевірка на відповідність вимогам	18.04.2025 – 30.04.2025	
8	Остаточне оформлення тексту, підготовка до подання та презентації	01.05.2025 – 09.05.2025	

Здобувач(ка) вищої освіти

_____ (підпис)

Віктор ШЕВЧУК

(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

_____ (підпис)

Валентина ДАНИЛЬЧЕНКО

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 51 стор., 15 рис., 2 табл., 60 джерел.

У цій роботі представлено результати дослідження, метою якого було створення інтелектуальної системи керування дорожнім рухом. Вона поєднує сучасні методи комп'ютерного зору та індукційного зондування. Актуальність впровадження таких технологій обумовлена збільшенням трафіку в містах, необхідністю зменшення заторів, поліпшенням екологічної ситуації та підвищенням безпеки дорожнього руху.

На тлі швидкого розвитку інформаційних технологій традиційні системи з фіксованими фазами світлофорів втрачають ефективність. Це зумовлює потребу в більш гнучких та адаптивних рішеннях, що враховують поточний стан транспортного потоку на основі актуальних даних. В рамках цієї роботи ми розробили програмно-апаратний комплекс, який дозволяє збирати, аналізувати та узагальнювати інформацію з відеокамер та індукційних петель з подальшим прийняттям рішень щодо керування світлофорами в реальному часі.

Система реалізована на базі компактного обчислювального пристрою Raspberry Pi 4, до якого підключено USB-камери та індукційні петлі. Для обробки зображень використано модель комп'ютерного зору YOLOv5, що забезпечує розпізнавання транспортних засобів з поділом на типи. Зважаючи на обмеженість ресурсів, характерну для вбудованих систем, особливу увагу приділено оптимізації алгоритмів, періодичному зчитуванню кадрів замість потокового відео, а також можливості гнучкої адаптації тривалості зеленого сигналу в залежності від дорожньої ситуації.

У процесі роботи були розглянуті сучасні підходи до управління дорожнім рухом у великих містах та проаналізовано функціональність провідних систем, таких як SCATS, SCOOT та Surtrac.

На основі даних з камер та петель система оцінює завантаженість доріг у кожному напрямку, визначає пріоритетність транспортних потоків та керує фазами світлофора відповідно до розрахованої логіки. Для наочності розроблено

інтерфейс, який відображає активні напрямки та фази в моделювальному середовищі, що дозволяє відстежувати поведінку системи в реальному часі. Важливим елементом є також розроблена логіка, що враховує не тільки кількість автомобілів, але й типи транспортних засобів, час очікування в кожному напрямку, потенційні конфлікти та навіть сценарії з обмеженою видимістю.

Модульна структура коду забезпечує гнучкість у налаштуванні, підтримку розширення (додавання нових камер або петель), а також можливість адаптації до реального обладнання, включаючи контролери світлофорів, які можуть підключатися через різні інтерфейси - UART, I²C, Ethernet або GPIO.

За допомогою імітаційних тестів було підтверджено, що комбінований підхід візуального аналізу та виявлення датчиків забезпечує більш точні дані про дорожній рух, ніж кожний з цих джерел окремо. Це дає змогу забезпечити більш справедливий розподіл фаз світлофорів, зменшити затримки та збільшити загальну пропускну здатність перехрестя. Отримані результати демонструють потенціал запропонованого рішення для використання в адаптивних системах управління дорожнім рухом, особливо в містах, де повноцінне впровадження дорогих комерційних систем є економічно недоцільним.

В подальшому система може бути вдосконалена для роботи з потоковим відео, розширення можливостей штучного інтелекту для прогнозування дорожніх ситуацій та інтеграції з іншими компонентами розумного міста, включаючи мобільні додатки, громадський транспорт та системи паркування.

Ключові слова: ТРАНСПОРТ, СВІТЛОФОР, ІНДУКЦІЙНА ПЕТЛЯ, КОМП'ЮТЕРНИЙ ЗІР, YOLOV5, RASPBERRY PI, ВІДЕОАНАЛІЗ, МУЛЬТИПЛЕКСОР, ІНТЕЛЕКТУАЛЬНА СИСТЕМА, ТРАФІК.

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

ПОДАННЯ ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ на здобуття освітнього ступеня бакалавра

Направляється здобувач Шевчук В.В. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 126
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)
на тему: «Система керування транспортним рухом з використанням технології комп'ютерного зору та індукційної петлі».

Кваліфікаційна робота і рецензія додаються.

Директор ННІ

(підпис)

Катерина НЕСТЕРЕНКО

(Ім'я, ПРІЗВИЩЕ)

Висновок керівника кваліфікаційної роботи

Здобувач Шевчук В.В. успішно виконав усі поставлені завдання, продемонстрував достатній рівень теоретичної обізнаності та практичної підготовки під час розробки системи керування транспортним рухом із застосуванням технологій комп'ютерного зору та індукційної петлі. Робота є актуальною, технічно обґрунтованою та повністю завершеною.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Шевчука В.В. на оцінку «відмінно» та присвоїти йому кваліфікацію бакалавр з інформаційних систем та технологій.

Керівник кваліфікаційної роботи _____

(підпис)

Валентина ДАНИЛЬЧЕНКО

(Ім'я, ПРІЗВИЩЕ)

«___» _____ 20__ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Шевчук В.В. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедру _____

(назва)

(підпис)

(Ім'я, ПРІЗВИЩЕ)

ВІДГУК РЕЦЕНЗЕНТА

на кваліфікаційну роботу

на здобуття освітнього ступеня бакалавра

здобувача вищої освіти Шевчук Віктор Володимирович

(прізвище, ім'я, по батькові)

на тему «Система керування транспортним рухом з використанням технології комп'ютерного зору та індукційної петлі»

Актуальність.

Тема кваліфікаційної роботи є надзвичайно актуальною в умовах зростаючої урбанізації та потреби в оптимізації транспортних потоків. Проблеми заторів, забруднення повітря та неефективного використання дорожньої інфраструктури вимагають впровадження сучасних інтелектуальних систем керування рухом. Використання комп'ютерного зору та індукційних петель є перспективним напрямком для створення ефективних та адаптивних рішень у цій галузі.

Позитивні сторони.

1. Тема роботи, що стосується оптимізації транспортного руху за допомогою сучасних технологій, розкрита всебічно, а всі поставлені завдання успішно реалізовані.
2. Орієнтація на практичне впровадження: Проєкт демонструє потенціал для автоматизації та підвищення ефективності управління дорожнім рухом, що має значну прикладну цінність.

Недоліки.

1. Обмеженість урізноманітнення сценаріїв тестування: Можливо, використані дані для моделювання або тестування не охоплюють достатньо широкого спектру реальних дорожніх ситуацій, що могло б розширити діапазон застосовності системи.
2. Можливість подальшого порівняльного аналізу: Хоча інтегровані дві технології, було б доцільно поглибити порівняльний аналіз їхньої ефективності в різних умовах, а також дослідити інші потенційні комбінації сенсорів чи алгоритмів для досягнення ще кращих результатів.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи бакалаврської.

Висновок: *кваліфікаційна робота на здобуття ступеня бакалавра заслуговує оцінку "відмінно", а здобувач Шевчук Віктор Володимирович заслуговує присвоєння кваліфікації: «бакалавр з інформаційних систем та технологій».*

Рецензент:

науковий ступінь, вчене звання

підпис

Ім'я, ПРІЗВИЩЕ

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ СУЧАСНОГО СТАНУ СИСТЕМ КЕРУВАННЯ ТРАНСПОРТНИМ РУХОМ.....	12
1.1 Формулювання мети та постановка завдань дослідження.....	12
1.2 Порівняльний аналіз алгоритмів обробки зображень.....	13
1.3 Аналіз технологій індукційного сенсування.....	18
1.4 Порівняння аналогів систем керування транспортним рухом.....	21
1.5 Обґрунтування вибору компонентів системи.....	25
1.6 Енергоефективність систем керування транспортним рухом.....	28
2 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ ТРАНСПОРТНИМ РУХОМ.....	33
2.1 Розробка архітектури програмної моделі системи керування.....	33
2.2 Розробка алгоритмів обробки даних із сенсорних пристроїв.....	34
2.3 Інтеграція програмного забезпечення з контролером світлофорів.....	36
3. МОДЕЛЮВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО РІШЕННЯ.....	38
3.1. Розробка програмного прототипу системи керування.....	38
3.2. Моделювання сценаріїв роботи системи та симуляція керування світлофорними режимами.....	45
3.3. Порівняльний аналіз ефективності програмного рішення з аналогами.....	49
3.4 Валідація результатів та аналіз похибок.....	53
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
Демонстраційні матеріали (презентація).....	67

ВСТУП

У сучасному світі, що відзначається бурхливим технологічним прогресом та безпрецедентним зростанням мобільності населення, питання ефективного управління транспортними потоками набуває критичного значення[1]. Мегаполіси та великі міста в усьому світі стикаються зі значними викликами, зумовленими постійним збільшенням заторів, зростанням кількості ДТП та нерівномірним розподілом транспортних потоків протягом доби. Ці проблеми не лише негативно впливають на економічну продуктивність міст, але й призводять до серйозних соціальних та екологічних наслідків, включаючи збільшення часу, витраченого на поїздки, підвищення рівня забруднення повітря та погіршення загальної якості життя міських мешканців.

Традиційні підходи до управління дорожнім рухом, які часто базуються на статичних алгоритмах та обмежених даних, виявляються недостатньо дієвими для вирішення цих складних проблем. Стає очевидною необхідність переходу до більш інтелектуальних та адаптивних систем управління[2]. У цьому контексті інтеграція інноваційних технологій, таких як комп'ютерний зір, штучний інтелект та сучасні сенсорні системи, у системи керування дорожнім рухом розглядається не просто як можливий варіант, а як нагальна потреба для забезпечення сталого розвитку міст та підвищення ефективності транспортної інфраструктури.

Дана кваліфікаційна робота присвячена розробці системи керування дорожнім рухом нового покоління, яка базується на комплексному використанні технологій комп'ютерного зору та індукційного контуру. Комп'ютерний зір, з його здатністю аналізувати відеопотоки, відкриває широкі можливості для автоматичного розпізнавання транспортних засобів, визначення їхніх типів, швидкості руху та напрямку, а також оцінки інтенсивності транспортного потоку. Водночас, індукційна петля, встановлена в дорожнє полотно, забезпечує надійні та точні дані про присутність транспортних засобів на конкретній ділянці дороги, що особливо важливо для виявлення заторів та оптимізації роботи світлофорів[3].

Поєднання цих двох технологій – комп'ютерного зору та індукційних петель – дозволяє створити більш повну та точну картину дорожньої ситуації, ніж кожна з них окремо. Це, в свою чергу, дозволяє розробляти більш ефективні алгоритми керування дорожнім рухом, які здатні оперативно реагувати на зміни інтенсивності руху, оптимізувати тривалість фаз світлофорів та зменшувати затримки транспортних засобів на перехрестях. Крім того, розроблена система може бути використана для збору цінних даних про транспортні потоки, які можуть бути використані для планування розвитку транспортної інфраструктури та прогнозування майбутніх потреб у транспортних послугах[4][5].

Основною метою цього дослідження є створення інноваційної та комплексної системи керування дорожнім рухом, що поєднує потужність сучасних методів обробки зображень та точність аналізу даних, отриманих від індукційних датчиків. Ця система спрямована на досягнення ключових цілей: підвищення рівня безпеки на дорогах, зменшення кількості дорожньо-транспортних пригод, оптимізація пропускної здатності транспортних магістралей та зменшення негативного впливу транспорту на навколишнє середовище шляхом скорочення викидів шкідливих речовин[6].

Для досягнення поставленої мети було виконано низку послідовних та взаємопов'язаних завдань, включаючи:

1. Проведення аналізу сучасних тенденцій та передового досвіду використання технологій комп'ютерного зору та індукційних систем у транспортному секторі, з метою виявлення найбільш перспективних підходів та визначення потенційних напрямків для інновацій.

2. Розробку детальної концепції інтегрованої системи управління дорожнім рухом, яка б поєднувала переваги обох технологій та забезпечувала ефективну взаємодію між ними[7].

3. Створення та оптимізацію алгоритмів аналізу даних, що дозволяють точно виявляти транспортні засоби, класифікувати їх за типами, вимірювати їхню швидкість та визначати інші важливі параметри руху.

4. Розробку програмного забезпечення для реалізації розроблених алгоритмів та інтеграції їх з наявною транспортною інфраструктурою.

5. Проведення тестування розробленої системи в умовах моделювання реальної дорожньої обстановки, з метою оцінки її ефективності, надійності та можливості масштабування[8].

Результати цієї роботи мають практичний потенціал та можуть бути використані для впровадження на різних рівнях – від окремих транспортних вузлів та перехресть до масштабних систем управління дорожнім рухом у цілих містах. Запропонована система сприяє не тільки оптимізації регулювання дорожнього руху та підвищенню його безпеки, але й забезпечує основу для розробки інтелектуальних транспортних систем майбутнього, які будуть здатні адаптуватися до постійно мінливих умов дорожнього руху та забезпечувати сталий розвиток міської мобільності. Таким чином, даний проект відкриває нові перспективи для розвитку підходів до управління транспортом, які повністю відповідають сучасним вимогам до інтелектуальних систем, характеризуються високою гнучкістю та масштабованістю, і мають значний потенціал для подальшого вдосконалення, розширення функціональності та адаптації до різноманітних умов експлуатації та потреб користувачів.

1 АНАЛІЗ СУЧАСНОГО СТАНУ СИСТЕМ КЕРУВАННЯ ТРАНСПОРТНИМ РУХОМ

1.1 Формулювання мети та постановка завдань дослідження

Метою даного дослідження є розробка концепції системи управління дорожнім рухом на основі інтеграції сучасних технологій комп'ютерного зору та сенсорної індукційної петлі[9]. Запропонований підхід передбачає використання апаратних засобів збору та обробки інформації, а саме встановлення відповідних датчиків, таких як камера та індукційна петля, для отримання даних з транспортного середовища[10]. Ці дані обробляються на вбудованій платі, яка, в свою чергу, керує роботою світлофорного контролера для оптимізації транспортного потоку. При цьому особлива увага приділяється розробці алгоритмів обробки інформації, які дозволять оперативно та точно реагувати на зміну умов руху та потенційні загрози безпеці.

Для досягнення мети дослідження були сформульовані наступні завдання. По-перше, необхідно провести детальний аналіз існуючих технологій комп'ютерного зору в транспортних системах, виявити їх переваги та недоліки з точки зору застосування в реальних умовах. По-друге, необхідно проаналізувати принципи роботи індукційної петлі як одного з основних датчиків, обґрунтувати вибір цього рішення та визначити його можливості при інтеграції в систему управління дорожнім рухом. По-третє, завданням дослідження є формулювання вимог до програмно-апаратного комплексу, здатного виконувати оперативну обробку сигналів від вбудованих датчиків та керувати світлофорними об'єктами на базі існуючого контролера.

Дослідження передбачає розробку алгоритмів, які забезпечать своєчасну обробку відеоінформації та даних з індукційного кола, а також інтеграцію отриманих результатів для формування сигналів керування. Ретельне вивчення питання включає аналіз сучасної літератури та практичних рішень, що дозволить удосконалити існуючі підходи до управління дорожнім рухом, сприятиме

підвищенню безпеки дорожнього руху та оптимізації інфраструктурних процесів. Таким чином, дослідження буде спрямоване на розробку інтегрованої моделі управління, що враховує сучасні технологічні тенденції та практичні вимоги до забезпечення ефективного функціонування транспортних систем.

1.2 Порівняльний аналіз алгоритмів обробки зображень

Сучасні технології комп'ютерного зору посідають провідне місце в розвитку інтелектуальних транспортних систем, які забезпечують автоматизований моніторинг та аналіз дорожньої обстановки. Застосування цієї технології дає змогу не тільки відстежувати поточний стан дорожнього руху, але й прогнозувати його динаміку, що, своєю чергою, сприяє оперативному реагуванню на зміни ситуації в режимі реального часу[11]. До ключових переваг використання комп'ютерного зору в транспортних системах можна зарахувати можливість реалізації автоматизованого виявлення транспортних засобів, аналізу їхніх характеристик (розмір, тип, швидкість), а також розпізнавання номерних знаків за допомогою сучасних алгоритмів машинного навчання та обробки зображень[12][13].

Однією з ключових особливостей комп'ютерного зору є здатність системи аналізувати великі потоки даних, які надходять як з відеокамер, так і з інших сенсорних пристроїв, розташованих вздовж великих автомагістралей та в містах. Це дає можливість не лише оперативно реагувати на надзвичайні ситуації, але й оптимізувати роботу світлофорів, що безпосередньо впливає на ефективність управління транспортними потоками та зменшує аварійність на дорогах[14].

Поглиблюючи аналіз, варто зазначити, що ефективність алгоритмів комп'ютерного зору в системах керування рухом значною мірою залежить від їхньої здатності обробляти та інтерпретувати великі обсяги даних, що надходять з різних джерел. Сучасні ІТС часто інтегрують дані не лише з відеокамер, але й з інших сенсорів, розташованих вздовж доріг, що дозволяє створити комплексну картину транспортної ситуації. Це, в свою чергу, вимагає від алгоритмів високої

пропускної спроможності та швидкодії, щоб забезпечити прийняття рішень у реальному часі[15][16].

Іншим важливим аспектом є адаптивність алгоритмів до змінних умов освітлення, погодних умов та інших чинників, що впливають на якість відеосигналу. Сучасні методи постобробки зображень, такі як нормалізація яскравості, підвищення контрастності та видалення шумів, у поєднанні з алгоритмами глибинного навчання дають змогу значно підвищити точність виявлення та класифікації об'єктів. Це, своєю чергою, забезпечує більш точне ухвалення рішень на основі аналізу отриманої інформації, зменшуючи ризик помилок під час визначення дорожньої обстановки.

Як наслідок, використання технологій комп'ютерного зору в інтелектуальних системах управління дорожнім рухом відкриває широкі можливості для впровадження інноваційних підходів до регулювання транспортних потоків. Це не тільки сприяє підвищенню ефективності та точності управління, але й істотно покращує загальний рівень безпеки дорожнього руху, що є важливим завданням у контексті зростаючих потреб сучасного суспільства в ефективних транспортних рішеннях.

Сучасні мережі, зокрема згорткові нейронні мережі (CNN), демонструють високу точність та ефективність у задачах класифікації, виявлення та сегментації об'єктів[17][18][19]. Серед популярних архітектур, які активно використовуються в системах моніторингу транспорту, є такі мережі, як YOLO (You Only Look Once), Faster R-CNN та SSD (Single Shot MultiBox Detector). Кожна з цих мереж має власні характеристики, що визначають можливості їх застосування в реальному часі, а також вимоги до обчислювальних ресурсів, що є критичним фактором при інтеграції у системи з обмеженими ресурсами.

Протокол YOLO (You Only Look Once) відзначається вражаючою швидкістю обробки зображень, що є вкрай важливим для додатків, котрі працюють в режимі реального часу, де кожна мілісекунда затримки може відігравати суттєву роль, як от в автоматизованих системах керування або автономних транспортних засобах[20]. Такий високий показник досягається завдяки тому, що всі розрахунки,

необхідні для знаходження об'єктів на зображенні, виконуються за один прохід через нейронну мережу, на відміну від інших підходів, які можуть вимагати кількох ітерацій або аналізу окремих частин зображення[21]. Головною перевагою такої архітектури є здатність обробляти все зображення як ціле, що не тільки мінімізує затримки в обробці відеопотоку, забезпечуючи практично миттєву реакцію на зміни в кадрі, але й полегшує апаратну реалізацію моделі, роблячи її придатною для використання на платформах з обмеженими обчислювальними ресурсами. YOLO відрізняється великою швидкістю обробки зображень, що є критичним для систем реального часу, оскільки всі обчислення, необхідні для визначення об'єктів, відбуваються всього за один прохід нейронною мережею. Ця особливість архітектури YOLO мінімізує затримки в обробці відео, забезпечуючи майже миттєву реакцію на зміни в кадрі. Основною перевагою такої продуктивності є можливість обробки всього зображення як єдиного цілого, на відміну від інших методів, що потребують декілька проходів або аналізу окремих частин. Цей цілісний підхід не лише зменшує обчислювальне навантаження, а й спрощує реалізацію моделі на платформах з обмеженими ресурсами[22].

Розглядаючи детальніше алгоритм YOLO, варто відзначити, що він розділяє вхідне зображення на фіксовану сітку, де кожна комірка цієї сітки відповідає за виявлення об'єктів, центри яких потрапляють у межі цієї комірки. Мережа одночасно прогнозує обмежувальні рамки та ймовірність належності об'єкта до певних класів, що значно скорочує час обробки у порівнянні з підходами, що спочатку виявляють потенційні області, а потім їх класифікують. Такий підхід дає можливість досягти високого рівня інтеграції з системами, де швидке прийняття рішень є критичним, наприклад, в автоматичних навігаційних системах для роботів або дронів. Але поділ зображення на фіксовану сітку може призвести до труднощів у розпізнаванні об'єктів, розташованих на межах комірок, або перекриття об'єктів в одній комірці, що може знизити точність виявлення у складних сценаріях.

Однак, незважаючи на значну ефективність у швидкості, YOLO має певні обмеження, особливо у точності виявлення об'єктів певних розмірів або у складних сценах. Зокрема, алгоритм може програвати у точності розпізнавання малих

об'єктів, що займають невелику площу на зображенні, оскільки інформації про них може не вистачити для надійної класифікації, або об'єктів, які розташовані на великій відстані від камери. Крім того, в умовах інтенсивного руху, коли об'єкти частково перекриваються, зливаються один з одним або мають складну просторову конфігурацію, YOLO може мати труднощі з їх чітким розрізненням, що призводить до зниження точності виявлення і збільшення кількості помилкових спрацьовувань.

На відміну від YOLO, Faster R-CNN (Регіональна згорткова нейронна мережа) орієнтована на досягнення максимальної точності виявлення об'єктів, що є критичним для багатьох задач, що вимагають високої надійності та деталізації, як-от медична діагностика або автономні навігаційні системи[23]. Це підхід базується на ідеї регіональних підказок, де мережа спершу використовує згорткову нейронну мережу (CNN) для вилучення ознак з усього зображення, створюючи карту ознак, котра відображає важливі характеристики зображення. Далі генерується набір регіональних пропозицій, які потенційно можуть містити об'єкти, використовуючи мережу регіональних пропозицій (RPN), яка ефективно сканує карту ознак для визначення областей інтересу. Кожна пропозиція піддається окремому аналізу, який визначає клас об'єкта та уточнює координати меж, забезпечуючи високу точність локалізації. Такий підхід забезпечує високу точність та дозволяє досягти значно кращої якості виявлення, особливо в завданнях, де пріоритетом є точна локалізація кожного об'єкта, визначення його меж та класифікація з високою достовірністю, але обчислювальна складність та обробка великої кількості регіональних даних потребує значних обчислювальних ресурсів, що небажано в умовах обмеженого обладнання або коли необхідно обробляти відео в режимі реального часу[24].

Проте, високий рівень точності, який забезпечує Faster R-CNN, неминуче супроводжується збільшенням часу обробки, оскільки кожна згенерована область потребує окремого етапу класифікації та уточнення меж. Таке збільшення обчислювального навантаження може стати серйозною проблемою при роботі з потоковими відеоданими в реальному часі, особливо на пристроях з обмеженими

обчислювальними можливостями, таких як вбудовані системи або мобільні платформи, де кожна мілісекунда затримки може вплинути на продуктивність системи.

Мережа SSD (Single Shot MultiBox Detector) є архітектурою виявлення об'єктів, що позиціонується між YOLO та Faster R-CNN, демонструючи оптимальний баланс між швидкістю обробки зображень та точністю виявлення[25]. На відміну від Faster R-CNN, SSD уникає окремого етапу генерування регіональних пропозицій, що суттєво прискорює процес обробки даних[26].

В основі SSD лежить принцип використання багатовимірних сіток для обчислення виявлень. Цей підхід дозволяє досягти високої швидкості обробки, порівнюваної з YOLO, але на відміну від YOLO, SSD демонструє кращу здатність розпізнавати об'єкти різного розміру та масштабу. Така особливість робить SSD надзвичайно корисною для сценаріїв, де вхідні дані можуть містити об'єкти значно відмінні за розміром, наприклад, під час аналізу дорожнього руху, коли потрібно виявляти як великі вантажівки, так і менші легкові автомобілі чи пішоходів.

Архітектура SSD базується на використанні кількох шарів ознак, отриманих з різних шарів згорткової нейронної мережі (CNN). Це забезпечує ефективне виявлення об'єктів різних розмірів, оскільки різні шари CNN здатні вловлювати інформацію про об'єкти на різних рівнях абстракції. Для забезпечення ефективного виявлення об'єктів різної форми та розміру, SSD використовує заздалегідь розраховані набори опорних блоків (anchor boxes або default boxes) на кожному шарі. Ці опорні блоки служать як початкові припущення щодо можливих розмірів та форм об'єктів, що дозволяє мережі одночасно передбачати кілька об'єктів з різними характеристиками[27].

Інтегрований підхід до обробки ознак, що поєднує інформацію з різних шарів CNN, сприяє підтримці високої точності виявлення. Одночасно оптимізована архітектура мережі, розроблена для мінімізації обчислювальних затрат, забезпечує скорочення часу обробки, що є критично важливим у динамічних сценаріях, таких як системи управління дорожнім рухом, системи відеоспостереження в реальному часі та інші додатки, де важлива швидка реакція.

1.3 Аналіз технологій індукційного сенсування

Індукційна петля є одним з найпоширеніших сенсорних компонентів, які використовуються в системах контролю дорожнього руху для виявлення наявності транспортних засобів на проїжджій частині. Головний принцип її роботи базується на явищі електромагнітної індукції, що дозволяє реєструвати зміни магнітного поля, спричинені появою в зоні дії датчика металевих предметів, в тому числі автомобілів.

Фізично індукційна петля являє собою провідник, зазвичай, мідний дріт, укладений у декілька витків, котрий вмонтований у дорожнє покриття (зазвичай у формі прямокутника). Цей провідник з'єднується з генератором високочастотного сигналу, утворюючи коливальний контур.

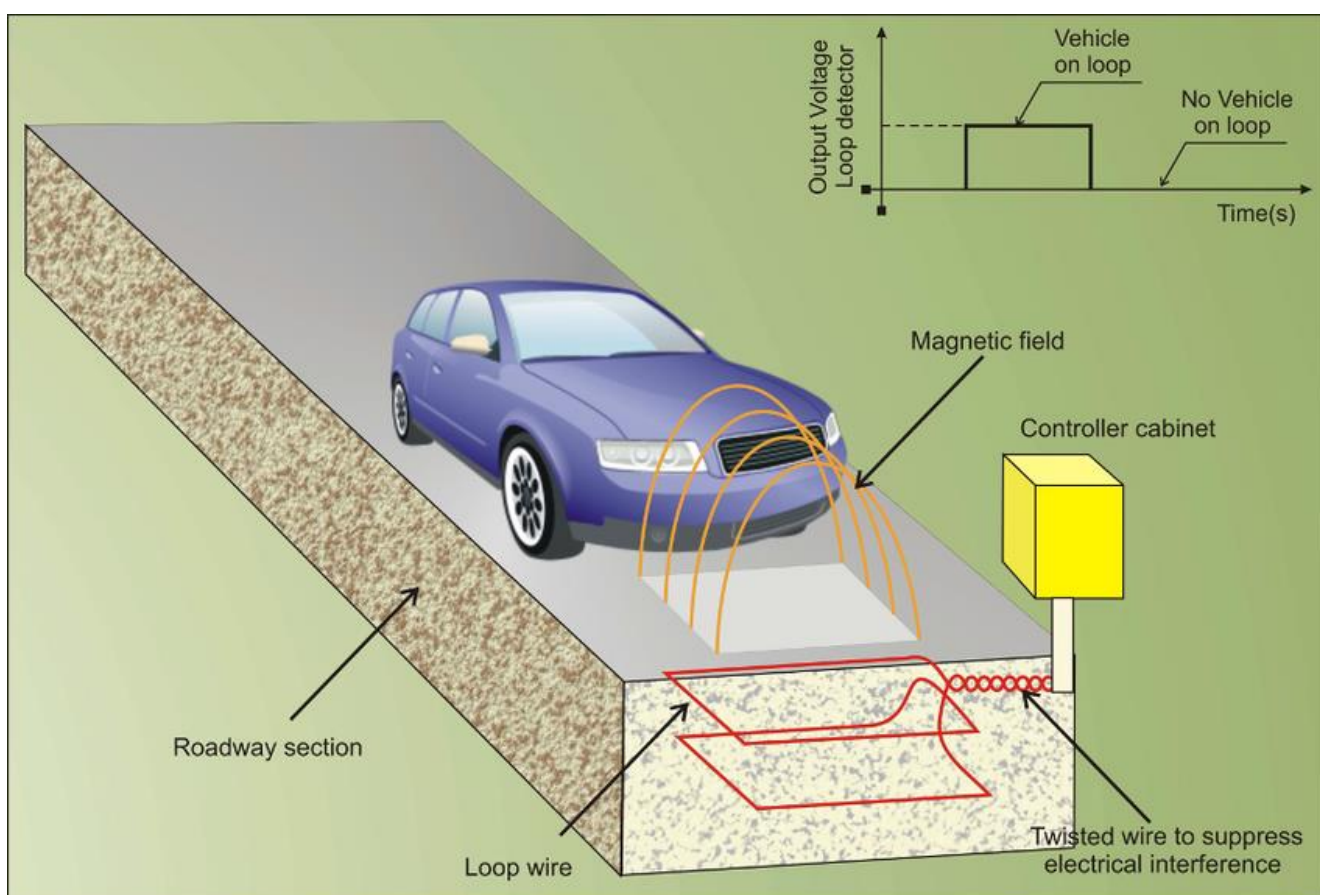


Рисунок 1.1-Принцип роботи індукційної петлі

Як видно на рисунку 1.1, петля утворює резонатор, в якому генератор підтримує змінний струм. Цей струм, циркулюючи по провіднику, створює навколо

нього магнітне поле, частота якого ретельно налаштовується. Коли металевий об'єкт, наприклад, автомобіль, в'їжджає в зону дії цього магнітного поля, відбувається цікаве явище. Метал кузова починає "поглинати" енергію магнітного поля, що індукує в ньому вихрові струми (так звані струми Фуко).

Це "поглинання" енергії не залишається безслідним для самої петлі – воно призводить до зміни її індуктивності. А зміна індуктивності, в свою чергу, викликає зсув резонансної частоти контуру. Саме цю зміну частоти й фіксує детектор, інтерпретуючи її як сигнал про наявність транспортного засобу.

Важливим аспектом є також конструкція та монтаж індукційних петель. Провід, з якого вона виготовляється, повинен мати надійну ізоляцію та бути стійким до агресивних факторів навколишнього середовища: вологи, перепадів температур, механічних пошкоджень. Спосіб укладання петлі в дорожнє полотно також має велике значення для її довговічності та ефективності. Зазвичай, в асфальті робиться спеціальний пропи́л, в який розміщується провід, а потім пропи́л герметизується.

Крім того, при проектуванні системи необхідно ретельно розраховувати розміри та форму петлі, а також кількість витків. Ці параметри визначаються конкретними завданнями, які має вирішувати система: виявлення легкових чи вантажних автомобілів, підрахунок трафіку, визначення напрямку руху тощо.

Варто також зазначити, що існують різні типи індукційних петель, які відрізняються своєю конфігурацією. Найпоширеніші форми – прямокутні, квадратні, але зустрічаються також круглі та діагональні петлі. Форма петлі впливає на її чутливість та зону виявлення.

Сучасні системи, що використовують індукційні петлі, часто виходять за рамки простого виявлення присутності. За допомогою кількох петель, розташованих на певній відстані одна від одної, можна реалізувати розширені функції, такі як визначення напрямку руху, вимірювання швидкості транспортних засобів та їх класифікація.

Цей процес дозволяє не лише фіксувати наявність транспортного засобу, а й, в залежності від конфігурації системи, підраховувати транспортні засоби,

визначати напрямок їхнього руху та навіть оцінювати розмір транспортного засобу для класифікації за типом (легковий, вантажний, мотоцикл тощо). Для досягнення такого рівня функціональності використовуються багатодієві шлейфи або декілька датчиків, встановлених в ряд, які дозволяють більш детально аналізувати просторові характеристики об'єкта, що перетинає індукційну зону.

Одними з важливих переваг індукційних петель є висока точність виявлення присутності транспортного засобу, надійність у складних погодних умовах і невимогливість до освітлення, що особливо важливо при використанні вночі або в умовах недостатньої видимості. Крім того, такі датчики мають тривалий термін служби і вимагають мінімального обслуговування після установки.

Проте, незважаючи на очевидні переваги, індукційні петлі мають ряд обмежень. Зокрема, процес монтажу є досить трудомістким і вимагає порушення дорожнього покриття, що ускладнює їхню інтеграцію в існуючі транспортні вузли. Крім того, індукційна петля може реагувати тільки на металеві предмети, що унеможливує виявлення пішоходів або велосипедистів без металевих елементів достатнього розміру. Крім того, через локалізований характер області зондування, петля виявляє тільки транспортні засоби, які зупинилися або проїхали безпосередньо над нею, що може створити труднощі при використанні на ширших ділянках дороги.

Не дивлячись на ці недоліки, індукційні петлі залишаються основним елементом багатьох сучасних систем виявлення транспортних засобів через їхню високу точність, надійність і відносно низьку вартість[28]. У поєднанні з іншими датчиками та аналітичними технологіями, такими як відеоаналітика або радарні датчики, вони забезпечують інтегрований підхід до моніторингу руху та ефективного ухвалення рішень в автоматизованому управлінні дорожнім рухом.

1.4 Порівняння аналогів систем керування транспортним рухом

Системи керування транспортним рухом відіграють ключову роль в організації ефективного та безпечного руху в сучасних містах, і серед різноманітних підходів виділяються декілька провідних[29].

Наприклад, SCATS (Система Координованого Адаптивного Управління Дорожнім Рухом Сіднея) – розробка з Австралії, яка адаптивно керує трафіком і здобула широке визнання та застосування у багатьох містах світу[30]. Ця система, що виникла в Сіднеї, ґрунтується на всебічному зборі даних про рух транспортних засобів в режимі реального часу. Для цього використовують індукційні петлі, що розміщені під проїжджою частиною, а також додаткові датчики, такі як відеокамери та радары, встановлені на перехрестях та вздовж доріг. Такий багатоаспектний збір інформації дозволяє системі оцінювати інтенсивність трафіку, знаходити затори, визначати швидкість руху машин та інші ключові параметри, які характеризують дорожню обстановку[31].

Зібрані дані використовуються для динамічного регулювання тривалості фаз роботи світлофорів, оптимізуючи їх функціонування таким чином, щоб зменшити затримки та забезпечити безперешкодний рух транспортних потоків. Такий адаптивний підхід, на відміну від статичних алгоритмів, що не реагують на зміни трафіку, демонструє високу адаптивність до змінних умов дорожнього руху. Система дозволяє централізовано керувати всією мережею світлофорів з одного центру, при цьому залишаючи можливість вносити локальні зміни на рівні окремих перехресть, враховуючи конкретні ситуації. Це допомагає істотно скоротити кількість заторів, збільшити пропускну здатність доріг та підвищити загальну ефективність транспортної системи. SCATS показує стабільну та надійну роботу навіть у великих мегаполісах з великою інтенсивністю руху, таких як Сідней, де її було створено, або Торонто, де її успішно впровадили.

Водночас важливо пам'ятати, що впровадження та подальша підтримка системи SCATS передбачає значні початкові та поточні витрати. Це пояснюється необхідністю встановлення великої кількості датчиків, прокладання кабельних

мереж та створення центрів управління. Крім того, SCATS потребує розгалуженої та надійної інфраструктури, що включає в себе стабільний канал зв'язку між усіма компонентами системи для безперебійного обміну даними та оперативного реагування на зміни дорожньої ситуації. Будь-які перебої в роботі мережі чи окремих датчиків можуть негативно вплинути на ефективність всієї системи.

Ще одна з ключових та поширених систем керування дорожнім рухом – це SCOOT (Split Cycle Offset Optimization Technique), що розроблена у Великобританії. Аналогічно до системи SCATS, SCOOT використовує індукційні петлі, розташовані на дорогах, для збирання даних про транспортні потоки. Це дозволяє отримувати інформацію про інтенсивність та характеристики руху в реальному часі. Однак, на відміну від SCATS, SCOOT зосереджується на динамічному регулюванні тривалості циклу світлофора[32]. Замість просто адаптувати фази світлофора, SCOOT активно змінює загальну тривалість циклу світлофора на перехрестях відповідно до змін інтенсивності руху.

Такий підхід надає системі SCOOT значно більшу гнучкість у налаштуванні режимів роботи окремих перехресть. Це дозволяє оптимізувати дорожній рух як на макрорівні, координуючи роботу великої кількості світлофорів у місті, так і на мікрорівні, враховуючи специфічні потреби кожного окремого перехрестя. SCOOT демонструє високу ефективність в умовах змішаного руху, коли на дорогах одночасно присутні різні види транспорту, включаючи приватні автомобілі, автобуси, тролейбуси та інші види громадського транспорту. Система здатна враховувати особливості руху кожного виду транспорту та оптимізувати цикли світлофорів таким чином, щоб забезпечити найефективніший розподіл часу між ними.

До того ж, SCOOT добре інтегрується з іншими системами керування міською інфраструктурою. Її можна поєднувати з системами моніторингу громадського транспорту, системами управління паркуванням та іншими інтелектуальними транспортними системами для створення комплексної системи керування мобільністю в місті.

Варто відзначити, що, незважаючи на численні переваги, система SCOOT має і певні недоліки. Процес її налаштування та калібрування досить складний і трудомісткий, вимагає залучення висококваліфікованих фахівців та значних початкових інвестицій. Для забезпечення ефективної роботи SCOOT необхідно постійно оновлювати картографічні дані міста, інформацію про дорожню мережу, зміни в організації дорожнього руху та інші актуальні дані, що також потребує значних ресурсів.

Альтернативний підхід до управління трафіком втілено в системі Surtrac (Scalable Urban Traffic Control), що була розроблена в Університеті Карнегі-Меллон, що у Сполучених Штатах Америки. На відміну від традиційних централізованих систем, Surtrac ґрунтується на принципах децентралізованого керування, де кожен світлофор автономно вирішує, як працювати, опираючись на локальні дані та аналізуючи ситуацію безпосередньо поруч[33]. Хоча й автономність, це не означає, що кожен світлофор працює ізольовано, оскільки передбачено механізми взаємодії між сусідніми світлофорами, що дає змогу загалом оптимізувати транспортні потоки в межах зони контролю.

Для ефективної реалізації децентралізованого управління, Surtrac застосовує передові технології, як-от комп'ютерний зір для аналізу відеопотоків з камер спостереження, аналіз даних з різних датчиків, встановлених на дорогах, а також методи машинного навчання для прогнозування транспортних потоків та оптимізації роботи світлофорів. Завдяки такому комплексному підходу, Surtrac здатна динамічно пристосовуватись до змінних умов дорожнього руху, забезпечуючи більш плавний та ефективний транспортний потік.

Ця архітектура системи забезпечує високу масштабованість, що є важливою перевагою для великих міст з розгалуженою дорожньою мережею, оскільки дозволяє легко розширювати зону контролю системи без значних змін у її структурі. Окрім цього, децентралізований підхід зменшує середній час очікування транспортних засобів на світлофорах, тому що кожен світлофор оптимізує свою роботу базуючись на локальній ситуації, а не на загальних усереднених показниках. Surtrac також демонструє високу здатність ефективно адаптуватися до місцевих

умов, таких як геометрія перехресть, інтенсивність руху у різний час доби та наявність громадського транспорту, зберігаючи при цьому загальну синхронізацію руху у всій зоні контролю.

З іншого боку, впровадження та експлуатація Surtrac пов'язані з певними труднощами. Система потребує встановлення відносно потужних обчислювальних модулів на кожному перехресті для забезпечення обробки даних у реальному часі та прийняття оптимальних рішень. Також ефективна робота Surtrac значною мірою залежить від стабільної та надійної мережевої інфраструктури, яка забезпечує обмін інформацією між світлофорами. Якщо мережа нестабільна або недоступна, продуктивність системи може знизитися. Іншою проблемою є складність діагностики та усунення несправностей у розподіленій системі, бо локалізація джерела проблеми може зайняти більше часу, ніж у централізованих системах.

В Україні також розробляються та використовуються системи керування дорожнім рухом. Прикладом може слугувати Автоматизована система керування дорожнім рухом міста Києва (АСКДР), яка являє собою централізовану систему управління світлофорами[34]. АСКДР об'єднує значну кількість світлофорних об'єктів та частково інтегрована з камерами відеоспостереження, GPS-трекерами та індукційними датчиками. Перевагами цієї системи є її сумісність з існуючою інфраструктурою, підтримка локальних режимів роботи світлофорів у разі втрати зв'язку з центром управління та відносно низькі вимоги до кваліфікації технічного персоналу. Однак, АСКДР має обмежену автоматизацію та адаптивність, не інтегрована з системами машинного зору, а на деяких вузлах використовується застаріле програмне забезпечення.

Кожна з розглянутих систем має свої сильні та слабкі сторони, і вибір конкретної системи залежить від багатьох факторів, включаючи особливості транспортної інфраструктури міста, інтенсивність трафіку, наявність фінансових ресурсів та вимоги до масштабованості та гнучкості системи управління.

1.5 Обґрунтування вибору компонентів системи

Розробка системи управління дорожнім рухом з застосуванням комп'ютерного зору вимагає ретельного добору апаратних та програмних складників. Потрібно забезпечити безперебійну роботу в умовах обмежених ресурсів, не перевищуючи запланований бюджет. Основним критерієм вибору є співвідношення між ціною компонентів, їхньою функціональністю та надійністю у реальних умовах експлуатації. Особливу увагу приділено сумісності пристроїв, простоті налаштування, енергоефективності, масштабованості та здатності працювати автономно.

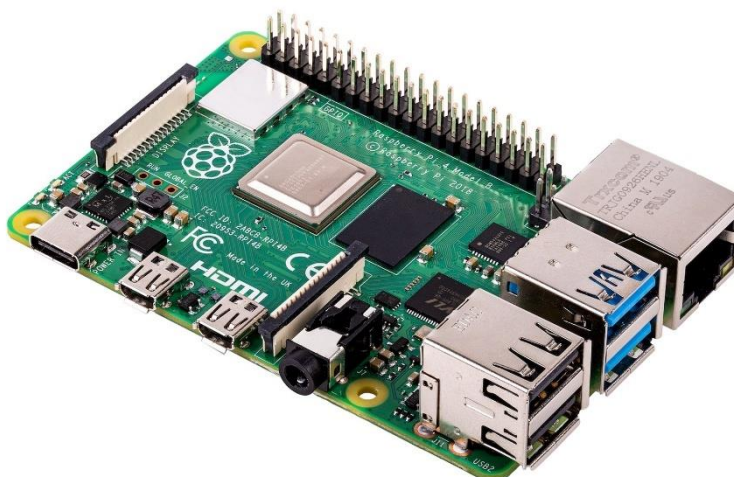


Рисунок 1.2-Плата Raspberry Pi 4 Model B

На рисунку 1.2 зображено обчислювальний модуль Raspberry Pi 4 Model B з 4 ГБ оперативної пам'яті, який було обрано для системи. Ця модель демонструє достатню обчислювальну потужність для роботи з нейронними мережами, що працюють з уже навченими моделями[35]. Також вона має підтримку USB 3.0 для швидкої обробки потоків з камер і забезпечує підключення до мережі через

Ethernet[36]. Це надзвичайно важливо для стабільної взаємодії з контролером світлофорів. Raspberry Pi - це доступний варіант з великою спільнотою підтримки, що полегшує розробку та усунення можливих проблем[37].



Рисунок 1.3-Камера Logitech C270

На рисунку 1.3 зображено веб-камеру Logitech C270 HD USB, яку планується використовувати для аналізу транспортного потоку. Це одна з найдоступніших веб-камер, що забезпечує прийнятну якість зображення (720p) при денному освітленні. Вона має надійні драйвери для Linux та легко інтегрується з Python, не вимагаючи складних налаштувань[38][39]. Хоча камера не підтримує нічну зйомку, цього цілком достатньо для денного аналізу в рамках завдання виявлення транспортних засобів на перехресті. У разі потреби, у майбутньому можна розглянути можливість заміни на моделі з ІЧ-підсвічуванням або CMOS-сенсором з вищою світлочутливістю.

У разі необхідності підключення більшої кількості камер, ніж наявних портів USB, можливо використати активні USB-хаби, що дозволяють масштабувати систему без втрати стабільності з'єднання. При цьому важливо враховувати пропускну здатність USB-шини та сумарне енергоспоживання[40].

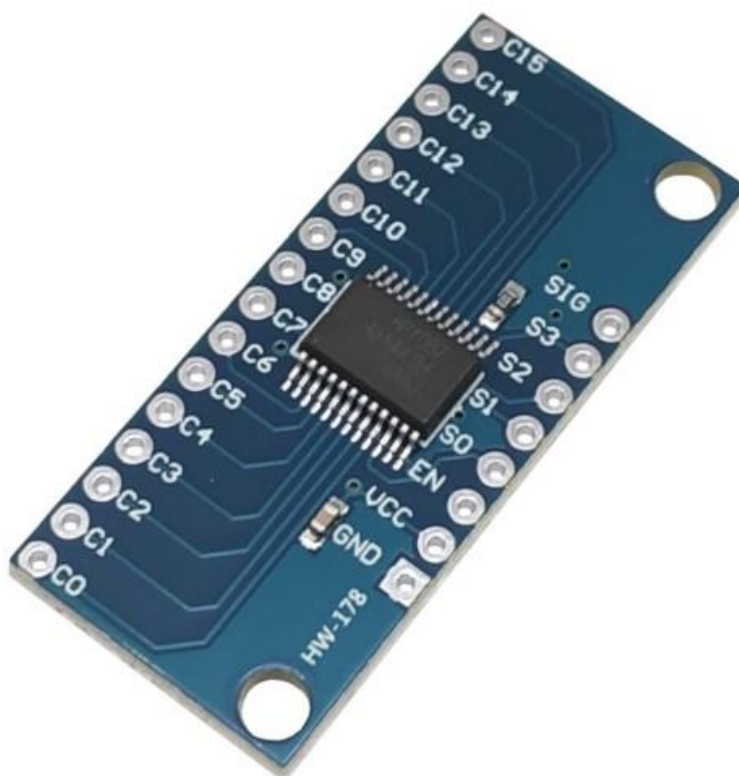


Рисунок 1.4-Мультиплексор CD74HC4067

Система передбачає збирання інформації про наявність транспортних засобів без візуального контакту. Для цього будуть використовуватися індукційні петлі які базовій конфігурації система передбачають пряме підключення через цифрові входи Raspberry Pi. Якщо ж кількість підключених петель перевищує доступну кількість GPIO або аналогових входів, можливе використання мультиплексорів, наприклад, CD74HC4067, зображеного на рисунку 4. Таке рішення дозволяє опитувати значно більшу кількість сенсорів, не ускладнюючи електронну частину системи[41]. Даний варіант значно дешевше, ніж окремі модулі зчитування для кожного контуру та забезпечує гнучкість при масштабуванні системи. Мультиплексори дають змогу опитувати сигнали від усіх індукційних датчиків по черзі з мінімальною затримкою[42].

Вибір індукційних датчиків був обумовлений їхньою стійкістю до погодних умов, простотою монтажу в дорожньому полотні та високою точністю виявлення металевих об'єктів. Обрано моделі на базі мікроконтролерів з дискретним виходом,

які при виявленні об'єкта змінюють логічний стан вихідного сигналу, який легко зчитується платою.

Таким чином, запропонований набір компонентів забезпечує оптимальний баланс між вартістю та функціональністю. Всі вибрані пристрої широко доступні, мають відкрите програмне забезпечення або драйвери, можуть бути легко замінені аналогами, а головне – дозволяють реалізувати адаптивну систему управління дорожнім рухом на локальному рівні, уникаючи необхідності використання дорогого серверного обладнання або спеціалізованих промислових контролерів.

1.6 Енергоефективність систем керування транспортним рухом

Підвищення ефективності споживання енергії в системах управління дорожнім рухом є ключовим аспектом оновлення міської інфраструктури, що не тільки знижує витрати на експлуатацію, але й позитивно впливає на довкілля. Дослідження цього питання включає оцінку теперішнього стану, аналіз впливу сучасних технологій, впровадження відновлюваних джерел енергії, оптимізацію компонентів, розробку відповідних показників та інтелектуальних стратегій управління енергоспоживанням.

Аналіз енергоспоживання традиційних світлофорних систем, що функціонують на базі фіксованих циклів перемикання сигналів, демонструє їх суттєве споживання енергії. Такі системи часто використовують застарілі лампові технології та споживають електроенергію без перерви, незалежно від фактичної інтенсивності руху, що веде до марної витрати ресурсів, особливо у періоди низького трафіку, наприклад, вночі. Нездатність адаптуватися до змінних умов є ключовою вадою, яка зумовлює неефективне використання енергії.

Вивчення впливу адаптивних систем на скорочення споживання енергії показує значний потенціал для економії. Адаптивні системи оптимізують тривалість роботи світлофорів відповідно до фактичного транспортного потоку[43]. Це не тільки покращує пропускну здатність перехрестя, а й опосередковано зменшує загальне енергоспоживання транспортної системи,

зменшуючи час простою двигуна в очікуванні зеленого сигналу та забезпечуючи більш плавний рух транспорту, що сприяє підвищенню паливної ефективності. Окрім цього, адаптивні системи самі можуть бути спроектовані таким чином, щоб оптимізувати власне енергоспоживання, активуючи периферійні пристрої або обчислювальні модулі лише тоді, коли це потрібно.

Важливим напрямком підвищення енергоефективності є застосування відновлюваних джерел енергії для живлення систем управління. Інтеграція сонячних панелей, а в окремих випадках і компактних вітрогенераторів, дозволяє забезпечити автономне або частково автономне енергопостачання світлофорних об'єктів, контролерів та сенсорних систем. Це не тільки зменшує залежність від централізованих електромереж та пов'язаних з цим витрат, а й суттєво знижує вуглецевий слід транспортної інфраструктури, сприяючи екологізації міського середовища.

Ключову роль відіграє оптимізація енергоспоживання компонентів самої системи управління. Перехід на світлофори зі світлодіодами (LED) замість традиційних ламп розжарювання може зменшити споживання електроенергії для світлофорного обладнання на 80-90%[44]. Також важливо обирати енергоефективні датчики, такі як сучасні камери з низьким споживанням для систем комп'ютерного зору, та оптимізовані індукційні петлі. Використання енергоефективних мікроконтролерів та обчислювальних платформ, таких як Raspberry Pi, як передбачено у цій роботі, також сприяє загальному зниженню енергоспоживання системи[45].

Для практичного оцінювання можливості живлення розробленої системи від сонячних панелей, розглянемо орієнтовний розрахунок енергоспоживання ключових компонентів та потрібної потужності автономної системи живлення[46]. За основу візьмемо обчислювальний модуль Raspberry Pi 4, чотири USB-камери Logitech C270 та мультиплексор CD74HC4067.

Максимальне енергоспоживання основних компонентів системи оцінюється таким чином:

- Процесорний модуль Raspberry Pi 4 (при максимальному навантаженні): приблизно 7,5 Вт.

- Чотири USB-камери Logitech C270 (в активному режимі): приблизно 6 Вт.

- Мультиплексор CD74HC4067 (під час перемикання каналів): приблизно 0,05 Вт.

Загальна максимальна потужність цих компонентів складає близько 13,55 Вт. Варто зазначити, що енергоспоживання контролерів світлофорів та контролерів індукційної петлі на цьому етапі не враховується, адже їхні точні характеристики потужності потребують додаткового з'ясування. Якщо вони мають значне споживання енергії, ці значення потрібно буде додати до загального розрахунку.

Враховуючи цілодобову роботу системи (24 години), добове споживання енергії становитиме 325,2 Вт*год/день.

Щоб забезпечити надійну роботу системи, потрібно враховувати резерв потужності та втрати системи. Запас на можливі пікові навантаження, поступове старіння компонентів та інші непередбачувані фактори вважається рівним 25%. Отже, скориговане добове споживання становить 406,5 Вт*год/день.

Окрім того, слід враховувати втрати енергії в системі акумулятор-зарядний пристрій, оскільки її ефективність рідко досягає 100%. Припускаючи загальний ККД 80% (що означає необхідність генерувати в 1,25 рази більше енергії), розрахункова кількість енергії, яку сонячні панелі повинні виробляти щодня, становить 508,25 Вт*год/день.

Для забезпечення автономної роботи системи в періоди без достатнього сонячного світла (наприклад, у хмарні дні) необхідно передбачити відповідну ємність акумулятора. Ми припускаємо, що кількість днів автономної роботи дорівнює трьом, що є типовим значенням для багатьох регіонів України для систем середньої критичності. Потрібна корисна ємність акумулятора становить 1524,375 Вт·год.

Обираючи акумулятор, важливо брати до уваги допустиму глибину розряду (ДГР), яка впливає на термін служби акумулятора. Для літій-залізо-фосфатних

(LiFePO₄) акумуляторів, які характеризуються тривалим терміном служби та можливістю глибокого розряду, ДГР може бути 80-90%. Якщо припустити ДГР 80%, загальна необхідна ємність акумулятора складає 1905,47 Вт*год.

Для практичного вибору акумулятора його ємність часто виражається в ампер-годинах (А*год) для певної напруги системи. Для сонячних систем звичайно використовується напруга 12 В або 24 В. Якщо напруга системи акумулятора становить 12 В, ємність становитиме 158,8 А*год при 12 В.

Потрібна потужність сонячних панелей залежить від середньої кількості пікових годин сонячного сяйва (PSH) для певного регіону. PSH показує еквівалентний час, протягом якого сонячне випромінювання має інтенсивність 1000 Вт/м². Для України цей показник варіюється: влітку він може складати 4-6 PSH, а взимку падає до 1-2 PSH. Щоб забезпечити цілорічну роботу системи, варто орієнтуватися на мінімальні зимові або середньорічні значення. Якщо припустити консервативне значення 2 PSH/день, потрібна потужність сонячної панелі становить 254,13 Вт. Відтак, рекомендується використовувати сонячну панель або комплект панелей загальною потужністю щонайменше 260 Вт. За більш оптимістичного значення PSH, приміром, 3 години (для літа чи менш консервативна оцінка), необхідна потужність панелей зменшиться до 169,4 Вт.

Отже, щоб забезпечити три дні повної автономності для базової конфігурації системи, яка містить Raspberry Pi, чотири камери та мультиплексор, згідно з оцінками, знадобиться акумулятор LiFePO₄ напругою 12 В та значною ємністю близько 158,8 А*год, що відповідає 1905,47 Вт*год накопиченої енергії. Для підзарядки цього акумулятора за дві пікові сонячні години на день (2 PSH/день) знадобиться сонячна панель загальною потужністю приблизно 260 Вт.

Проте, при розробці реальної системи автономного живлення необхідно ретельно проаналізувати низку ключових аспектів. Перш за все, слід точно визначити повне енергоспоживання всієї системи, враховуючи не лише базову конфігурацію, але й будь-яке інше підключене обладнання, таке як контролери світлофорів, датчики індукційних петель та інша периферія. Значне

енергоспоживання цих додаткових компонентів призведе до пропорційного збільшення необхідної ємності акумуляторів та потужності сонячних панелей.

Критично важливим є точне визначення значення пікових сонячних годин (PSH) для конкретної локації. Цей параметр має значні коливання залежно від географічного розташування об'єкта, пори року, кута нахилу та орієнтації сонячних панелей, а також наявності будь-яких факторів затінення. Для забезпечення надійної роботи системи рекомендується використовувати детальні кліматичні дані, специфічні для місця встановлення.

Окрім того, для коректної роботи автономної системи живлення знадобиться додаткове обладнання. Обов'язковим елементом є сонячний контролер заряду, причому для досягнення максимальної ефективності зарядки акумулятора від сонячних панелей рекомендується використовувати контролер типу MPPT, який оптимізує процес зарядки, запобігає пошкодженню акумулятора та збільшує кількість отриманої енергії. Також, оскільки Raspberry Pi та деякі інші компоненти системи працюють від напруги 5В, а акумуляторна система має напругу 12В, буде потрібен якісний DC-DC понижуючий перетворювач для стабільного живлення цих пристроїв.

Нарешті, слід враховувати вплив температурних умов на роботу системи. Ефективність перетворення сонячної енергії панелями та ємність акумуляторних батарей можуть суттєво змінюватися залежно від температури навколишнього середовища. Ці температурні залежності необхідно брати до уваги при виборі обладнання та проектуванні системи для забезпечення її надійної роботи в різних кліматичних умовах.

Наведені розрахунки є попередньою оцінкою і слугують для демонстрації принципової можливості створення автономної системи живлення. Для розробки конкретного проекту рекомендується проведення більш детальних розрахунків з урахуванням усіх специфічних умов експлуатації та, за потреби, консультація з фахівцями в галузі сонячної енергетики.

2 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ ТРАНСПОРТНИМ РУХОМ

2.1 Розробка архітектури програмної моделі системи керування

Для ефективного керування дорожнім рухом потрібна програмна архітектура, котра буде збирати дані з різних джерел, оперативно реагувати на зміни на дорозі та гарантувати стабільний зв'язок з пристроями управління світлофорами. Ключовим є гнучкість системи, можливість масштабування та відносна простота впровадження на різних апаратних платформах, враховуючи навіть обмежені ресурси.

Запропонована система побудована на модульній архітектурі, що складається з трьох основних функціональних блоків: модулю збору даних, модулю обробки інформації та прийняття рішень і модулю взаємодії зі світлофорним контролером.

Модуль збору даних об'єднує інформацію з камер відеоспостереження та індукційних петель. Індукційні петлі використовуються для точного виявлення наявності транспортних засобів. Ці пристрої відрізняються високою точністю, швидкістю реакції та мінімальною кількістю помилкових спрацьовувань. Сигнали з індукційних петель передаються на обчислювальний пристрій у вигляді цифрових сигналів, котрі відповідають на зміни магнітного поля, що викликаються проїздом автомобілів.

Для кожного напрямку руху до перехрестя встановлено камеру, яка підключена до обчислювального пристрою. З метою ефективного використання обчислювальних ресурсів обробка відеопотоку не є безперервною. Натомість, система періодично отримує окремі кадри (зображення) з камер. Частота отримання кадрів може змінюватися.

Модуль обробки інформації та прийняття рішень відповідає за аналіз зібраних даних. Обробка зображень з камер відбувається за допомогою моделі YOLOv5s, що забезпечує швидке та доволі точне розпізнавання об'єктів[47][48][49]. YOLOv5s здатна розпізнавати різні види транспорту (легкові

авто, вантажівки, автобуси) та пішоходів. На основі інформації про кількість транспортних засобів у кожному напрямку, а також даних з індукційних петель, система обчислює оптимальну тривалість зеленого сигналу світлофора для кожного потоку руху.

Модуль взаємодії зі світлофорним контролером надсилає згенеровані системою команди на пристрій, що безпосередньо керує сигналами світлофора. Спосіб передачі команд залежить від типу використовуваного контролера світлофора і може здійснюватися через різні інтерфейси, такі як GPIO[50], UART[51][52] або I2C[53]. Для забезпечення коректної роботи системи застосовується буфер обміну для координації сигналів та уникнення конфліктів при зміні фаз світлофора.

Загалом, ця модульна архітектура забезпечує необхідну гнучкість, масштабованість та ефективність системи управління дорожнім рухом, дозволяючи обробляти дані з різних джерел, приймати обґрунтовані рішення та ефективно керувати світлофорами.

2.2 Розробка алгоритмів обробки даних із сенсорних пристроїв

У системі управління дорожнім рухом, що базується на даних з камер та індукційних петель, надзвичайно важливо запровадити ефективні алгоритми збору та обробки даних. Дані з датчиків є основою для формування адаптивної логіки світлофорів, тому їх опрацювання повинне бути оптимізоване як за часом, так і за обчислювальними ресурсами[54].

Алгоритм опрацювання відеоінформації з камери.

Зображення з камер надходять на обчислювальну платформу (Raspberry Pi) через фіксований інтервал часу. Такий підхід дозволяє уникнути перевантаження системи, забезпечуючи при цьому достатньо часте оновлення ситуації на перехресті.

Обробка одного зображення відбувається в такій послідовності:

1. Захоплення кадру з потоку відповідної камери.

2. Попередня обробка зображення: зміна розміру, нормалізація значень пікселів, перетворення у формат, що відповідає вхідним вимогам моделі YOLOv5s.

3. Передача зображення нейронній мережі, яка виявляє транспортні засоби на зображенні.

4. Агрегація результатів - підрахунок кількості транспортних засобів у кожному напрямку наближення до перехрестя.

5. Збереження даних у буфері та передача результатів до модуля прийняття рішень.

Особливістю реалізації є можливість визначення щільності руху не тільки за кількістю транспортних засобів, але й за їхньою типологією. Наприклад, наявність великогабаритного транспорту може впливати на тривалість фази зеленого сигналу. Крім того, результати з декількох камер об'єднуються для побудови повної картини завантаженості перехрестя.

Алгоритм аналізу сигналів з індукційної петлі.

Індукційні петлі відіграють роль базового детектора транспортних засобів. Вони надають просту, але надійну інформацію про наявність транспортного засобу на певному напрямку руху. Алгоритм обробки цих сигналів значно простіший, ніж опрацювання зображень, але не менш важливий.

Основні етапи обробки сигналів:

1. Моніторинг вхідного сигналу з контуру в реальному часі.
2. Фіксація подій появи транспортного засобу (зміни стану сигналу).
3. Ведення журналів активності по кожному шлейфу, що дозволяє враховувати не тільки наявність транспортного засобу, але і тривалість його очікування.
4. Об'єднання інформації з декількох петель для кожного напрямку, що дозволяє оцінити завантаженість напрямків руху окремо.

Наявність індукційної петлі на кожній смузі дозволяє також реалізувати функцію виявлення незавантажених напрямків, що є ключовим для мінімізації простою світлофора в порожньому циклі.

Узгодження результатів та подальша обробка.

Дані з камер і шлейфів об'єднуються в загальну таблицю станів, яка передається в модуль прийняття рішень. У разі суперечливих даних (наприклад, камера не виявила автомобіль, а шлейф спрацював) система застосовує пріоритетну обробку, віддаючи перевагу індукційному датчику як більш надійному джерелу в умовах обмеженої видимості (дощ, туман, затемнення тощо).

Результатом роботи вищевказаних алгоритмів є поточна оцінка трафіку для кожного напрямку перехрестя який має наступну інформацію: кількість транспортних засобів, наявність вантажівок, час очікування та вагові коефіцієнти для прийняття рішень. Ці дані стають основою для алгоритмічного визначення наступної фази світлофорного циклу.

2.3 Інтеграція програмного забезпечення з контролером світлофорів

Для ефективної роботи інтелектуальної системи управління трафіком потрібен чітко налагоджений механізм передачі інструкцій на світлофорне устаткування. Тому ключовим етапом проектування є створення інтерфейсу взаємодії між обчислювальним модулем, у ролі якого виступає Raspberry Pi, та контролером світлофора. Цей інтерфейс мусить забезпечувати ясну, оперативну й безпечну передачу інформації, яка визначає режими регулювання світлофорів відповідно до поточної ситуації на дорогах, оціненої за допомогою комп'ютерного зору та даних з індукційних датчиків.

Зважаючи на відсутність стандартизованої документації у вільному доступі для більшості промислових контролерів світлофорів, підхід до розробки інтерфейсу слід зробити максимально гнучким та адаптивним. Передбачається, що взаємодія може відбуватися або через цифрові виходи Raspberry Pi (GPIO), або через один з класичних протоколів передачі даних – UART, I²C, SPI[55][56] чи навіть Ethernet[57][58], в залежності від доступного комунікаційного інтерфейсу на контролері. Такий підхід дозволяє легко адаптувати систему до різних моделей обладнання, мінімізуючи ризики виникнення проблем з сумісністю при подальшому розгортанні у реальних умовах[59].

В основі інтерфейсу лежить принцип передачі простих команд, які визначають фазу циклу світлофора: наприклад, увімкнути зелене світло для одного напрямку та одночасно червоне для перпендикулярного. Для цього розробляється внутрішній протокол обміну даними, що визначає формат команд (наприклад, JSON або бінарну структуру), логіку їх послідовності, правила підтвердження отримання, а також механізми виявлення й обробки помилок. У випадку реалізації інтерфейсу через протокол UART, команда може мати вигляд текстового рядка з унікальним ідентифікатором фази та тривалістю її дії.

Не менш важливим є врахування питань безпеки. В умовах реального міського середовища, де втручання в роботу світлофора може призвести до негативних наслідків, критично важливо забезпечити базові методи аутентифікації та захисту від несанкціонованого доступу. Це можливо реалізувати, обмеживши фізичний доступ до системи, або за допомогою логічного шифрування чи перевірки цифрового підпису переданих команд.

Створений інтерфейс необхідно ретельно тестувати в умовах, максимально наближених до реальних, імітуючи різні дорожні ситуації. Під час тестування потрібно перевірити коректність передачі команд, продуктивність системи, її стійкість до затримок або втрати сигналу, а також здатність відновлюватися після виникнення помилок. Якщо підключення до реального контролера неможливе, на етапі проектування допускається використання програмного емулятора або логічного аналізатора, що дасть змогу оцінити поведінку системи при зміні параметрів вхідних даних.

З огляду на вищезазначені аспекти, взаємодія між обчислювальним модулем і контролером світлофора є ключовим елементом загальної архітектури системи управління. Вона має бути не тільки технічно реалізованою, а й адаптивною до змін навколишнього середовища та різноманітного обладнання, що гарантує надійність та масштабованість системи при її впровадженні в реальному міському середовищі.

3. МОДЕЛЮВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО РІШЕННЯ

3.1. Розробка програмного прототипу системи керування

Центральним вузлом системи керування є головний скрипт, що запускає безперервний цикл моніторингу трафіку.

```
import time
import cv2
import RPi.GPIO as GPIO

from config import *
from logic import analyze_traffic
from controller import run_phase

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

for pin in MUX_ADDRESS_PINS + MUX_ENABLE_PINS:
    GPIO.setup(pin, GPIO.OUT)
for pin in MUX_ENABLE_PINS:
    GPIO.output(pin, GPIO.HIGH)
GPIO.setup(MUX_INPUT_PIN, GPIO.IN)

cams = [cv2.VideoCapture(i) for i in CAMERA_INDEXES]
```

Рисунок 3.1-Фрагмент коду ініціалізації системи

На рисунку 3.1 представлено фрагмент коду початкових налаштувань, необхідних для реалізації системи керування транспортним рухом. Він показує, як система готується до отримання даних з камер та взаємодії з мультиплексором для вибору потрібної камери. Цей крок є критичним для забезпечення коректної роботи системи, оскільки він визначає, як програмне забезпечення буде взаємодіяти з апаратним забезпеченням.

```

def select_mux_channel(channel: int):
    mux_id = channel // NUM_CHANNELS_PER_MUX
    channel_in_mux = channel % NUM_CHANNELS_PER_MUX

    for i in range(4):
        GPIO.output(MUX_ADDRESS_PINS[i], (channel_in_mux >> i) & 1)

    for i, en_pin in enumerate(MUX_ENABLE_PINS):
        GPIO.output(en_pin, GPIO.LOW if i == mux_id else GPIO.HIGH)

    time.sleep(0.01)

def read_loop_channel(channel: int) -> bool:
    select_mux_channel(channel)
    return GPIO.input(MUX_INPUT_PIN) == GPIO.HIGH

```

Рисунок 3.2-Реалізація вибору каналу для зчитування даних з індукційних петель

Рисунок 3.2 ілюструє, як система вибирає конкретну індукційну петлю для отримання даних. Функції `select_mux_channel` та `read_loop_channel` забезпечують цей процес. Важливо відзначити використання бітових операцій у `select_mux_channel`, які дозволяють ефективно декодувати номер каналу та встановлювати відповідні сигнали на адресних пінах мультиплексора. Це є ключовим для коректної роботи системи, оскільки забезпечує правильне зчитування даних з кожної індукційної петлі.

```

try:
    while True:
        for i in range(TOTAL_LOOPS):
            print(f"\n=== Перевірка напрямку {i} ===")

            loop_triggered = read_loop_channel(i)

            ret, frame = cams[i].read()
            if not ret:
                print(f"[!] Камера {i} не читає")
                continue

            result = analyze_traffic(frame, loop_triggered)

            print(f" ТЗ: {result['vehicle_count']} | Повільні: {result['slow_vehicle_count']}")
            print(f" Колір: {result['color']} | Час: {result['green_time']} с")

            if result["color"] == "green":
                run_phase(i, result["green_time"])

except KeyboardInterrupt:
    print("\n[!] Зупинка")
finally:
    for cam in cams:
        cam.release()
    GPIO.cleanup()

```

Рисунок 3.3-Головний цикл системи

На рисунку 3.3 зображено головний цикл системи керування транспортним рухом. Цей цикл – основна частина системи, яка забезпечує постійну роботу та реагування на зміни у дорожньому потоці. На кожній ітерації цього циклу система по черзі перевіряє кожен напрямок руху, що контролюється зв'язкою: індукційною петлею та відеокамерою. Мультиплексор, який перемикає канали, використовується для вибору активного напрямку. Адресна шина мультиплексора визначає, який канал (і відповідно напрямок) опитується на поточний момент. Стан індукційної петлі, що вказує на наявність чи відсутність транспорту, зчитується через спільний цифровий вхід мікроконтролера.

Водночас зчитується відеопотік з відповідної камери, надаючи візуальну інформацію про ситуацію на дорозі. Після збору цих даних – відеозображень та показів індукційної петлі – вони надсилаються модулю аналізу трафіку для подальшої обробки та ухвалення рішень. У випадку виявлення транспортного засобу, незалежно від способу – за допомогою петлі чи відеокамери, система активує процес ухвалення рішення про дозвіл руху в цьому напрямку. Це рішення

базується на комплексному аналізі отриманої інформації, що забезпечує адаптивне та ефективне керування трафіком.

```
import torch
import numpy as np
from config import WEIGHTS, MIN_GREEN, MAX_GREEN, SECONDS_PER_WEIGHT_UNIT

# Завантаження моделі
model = torch.hub.load('ultralytics/yolov5', 'custom', path='yolov5s.pt', force_reload=True)
model.conf = 0.5

TRANSPORT_CLASSES = ['car', 'motorcycle', 'bus', 'truck', 'bicycle']
SLOW_CLASSES = ['truck', 'bus']
```

Рисунок 3.4-Ініціалізація компонентів системи розпізнавання транспортних засобів

```
def process_frame(frame: np.ndarray):
    results = model(frame)
    detections = results.pandas().xyxy[0]
    return detections

def analyze_traffic(image: np.ndarray, loop_triggered: bool):
    detections = process_frame(image)
    transport_detections = detections[detections['name'].isin(TRANSPORT_CLASSES)]

    total = len(transport_detections)
    slow = sum(transport_detections['name'].isin(SLOW_CLASSES))

    only_one_detected = (loop_triggered and total == 0) or (not loop_triggered and total > 0)

    if only_one_detected:
        return {
            "vehicle_count": total,
            "slow_vehicle_count": slow,
            "green_time": MIN_GREEN,
            "color": "green"
        }

    duration = calculate_green_duration(total, slow)
    return {
        "vehicle_count": total,
        "slow_vehicle_count": slow,
        "green_time": duration,
        "color": "green" if total > 0 or loop_triggered else "red"
    }
```

Рисунок 3.5-Обробка фотоінформації та аналіз транспортного навантаження.

Обробка зображень є критичним етапом роботи системи і виконується в окремому логічному модулі, що ілюструє рисунок 3.5. Цей модуль застосовує

потужну нейронну мережу, навчену на великому масиві даних, для точного розпізнавання різних типів транспортних засобів. Модель може ідентифікувати легкові авто, мотоцикли, автобуси, вантажівки та навіть велосипеди. Рисунок 3.4 демонструє початкові налаштування, необхідні для ініціалізації системи розпізнавання транспортних засобів.

На основі кількості та класифікації розпізнаних об'єктів система розраховує оптимальний час увімкнення зеленого світла для конкретного напрямку. При цьому враховується ваговий коефіцієнт для кожного виду транспортного засобу: автомобілям, мотоциклам та велосипедам, як правило, що рухаються швидше, присвоюється вага 1, тоді як вантажівкам та автобусам, які мають більші габарити та меншу маневреність, дається вага 3.

Паралельно з аналізом відеоданих аналізується стан індукційної петлі, що слугує додатковим підтвердженням наявності транспорту. Якщо на зображенні або за допомогою індукційної петлі виявлено транспортний засіб, система генерує команду на активацію зеленої фази світлофора. Тривалість цієї фази динамічно визначається пропорційно до сумарної ваги транспорту, але не перевищує мінімальних та максимальних значень, зазначених у файлі конфігурації. Такий підхід дозволяє системі гнучко реагувати на поточну інтенсивність трафіку, водночас забезпечуючи безпеку та ефективність регулювання. Це означає, що система здатна віддавати пріоритет напрямкам з більшою кількістю транспорту, але при цьому уникає надмірно тривалих затримок для інших напрямків.

```

import socket
import json

# IP та порт контролера світлофорів
CONTROLLER_IP = "192.168.0.100"
CONTROLLER_PORT = 5000

def send_traffic_light_command(direction, color):
    data = {
        "direction": direction,
        "state": color
    }
    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.connect((CONTROLLER_IP, CONTROLLER_PORT))
            s.sendall(json.dumps(data).encode())
    except Exception as e:
        print(f"[!] Не вдалося відправити команду: {e}")

def run_phase(direction, green_time):
    print(f"[>] Ввімкнення зеленого світла для напрямку {direction} на {green_time} с")
    send_traffic_light_command(direction, "green")
    time.sleep(green_time)
    send_traffic_light_command(direction, "yellow")
    time.sleep(2)
    send_traffic_light_command(direction, "red")

```

Рисунок 3.6-Реалізація логіки перемикання фаз світлофора

Після того, як логічний модуль приймає рішення про активацію зеленої фази світлофора, головний скрипт надсилає відповідну команду керування до модуля керування світлофором. Цей модуль відповідає за взаємодію з фізичним контролером світлофора, як показано на рисунку 3.6, де представлено фрагмент коду, що реалізує цю функціональність. Для цього він генерує повідомлення у стандартному форматі JSON, що містить інформацію про напрямок, для якого слід увімкнути зелений сигнал, а також тривалість цієї фази. Згенероване повідомлення надсилається мережею Ethernet на IP-адресу контролера світлофора. Після завершення зеленої фази модуль керування автоматично генерує та надсилає команди для перемикання сигналу спочатку на жовтий (для попередження водіїв), а потім на червоний (для зупинки руху). Цей процес забезпечує чітку та послідовну зміну сигналів світлофора, що критично важливо для безпеки дорожнього руху.

Таким чином, система автоматично завершує цикл керування для цього напрямку та переходить до обробки наступного. Загалом, така схема забезпечує високу гнучкість в управлінні світлофорами, дозволяючи системі ефективно реагувати на різні дорожні ситуації, а також гарантує дотримання всіх необхідних вимог безпеки дорожнього руху. Застосування стандартизованого формату JSON та мережі Ethernet полегшує інтеграцію системи з наявними контролерами світлофорів та забезпечує її потенційне розширення в майбутньому.

```
MUX_ADDRESS_PINS = [4, 17, 27, 22]
MUX_ENABLE_PINS = [5]
MUX_INPUT_PIN = 18

NUM_CHANNELS_PER_MUX = 16
TOTAL_LOOPS = NUM_CHANNELS_PER_MUX * len(MUX_ENABLE_PINS)

CAMERA_INDEXES = list(range(TOTAL_LOOPS))
MIN_GREEN = 2
MAX_GREEN = 65
SECONDS_PER_WEIGHT_UNIT = 2
WEIGHTS = {"light": 1, "heavy": 3}
```

Рисунок 3.7-Конфігурації системи

Для забезпечення простоти налаштування та зручності обслуговування, всі системні параметри, такі як кількість виходів GPIO (загального призначення вхід/вихід) мікроконтролера, адреси мультиплексорів, параметри тривалості фаз світлофора (мінімальна та максимальна тривалість зеленого сигналу, тривалість жовтого сигналу), IP-адреса контролера світлофора, а також інші важливі змінні, зберігаються в окремому конфігураційному файлі, приклад якого наведено на рисунку 3.7. Це значно спрощує процес адаптації системи до різних умов експлуатації, наприклад, при зміні кількості камер або індукційних петель, без потреби вносити зміни до програмного коду самої системи. Такий підхід робить систему більш модульною та простою в обслуговуванні, що є важливим фактором при її розгортанні та експлуатації. Конфігураційний файл слугує єдиним місцем

для внесення змін, що знижує ймовірність помилок та спрощує процес оновлення системи.

Особлива увага при розробці системи була приділена обробці можливих помилкових або непередбачуваних ситуацій, які можуть виникнути під час її роботи. До таких ситуацій відносяться, наприклад, тимчасова відсутність сигналу з камери, короткочасне зникнення зображення (наприклад, через перешкоди) або збій живлення в індукційному контурі. Для забезпечення надійності системи розроблено механізми обробки цих ситуацій, що запобігають зупинці головного циклу системи. У разі виникнення проблеми з конкретним напрямком руху система автоматично пропускає його та переходить до опитування наступного, таким чином забезпечуючи безперервність керування рухом. Це критично важлива характеристика системи, оскільки забезпечує її стійкість до збоїв та гарантує, що рух буде регулюватися навіть в умовах тимчасових проблем. Такий підхід суттєво підвищує надійність системи та зменшує потребу втручання оператора для її відновлення після збоїв.

3.2 Моделювання сценаріїв роботи системи та симуляція керування світлофорними режимами.

У процесі моделювання керування транспортними потоками була реалізована імітаційна система, яка опрацьовує зображення з певної директорії, обробляє їх за допомогою моделі YOLOv5, і приймає рішення щодо тривалості зеленої фази світлофора залежно від кількості транспортних засобів на кожному напрямку. Результати моделювання аналізувалися шляхом виводу діагностичної інформації безпосередньо в консоль та окреме вікно візуалізації світлофорних режимів.



Рисунок 3.8-Аналіз імітаційного зображення

Рисунок 3.8 ілюструє приклад аналізу імітаційного зображення транспортного потоку. Перш за все, для кожного напрямку здійснюється

формування вхідного набору зображень. Кількість таких зображень вибирається випадково в межах від 0 до 30, що імітує різну транспортну інтенсивність у різні моменти часу. Цей підхід дозволяє протестувати систему в умовах як повної відсутності транспорту, так і при високому навантаженні. У виводі чітко фіксується кількість відібраних зображень для поточного напрямку, що дозволяє відразу оцінити обсяг даних, що надходить на обробку.

У випадках, коли зображення відсутні (тобто система отримує нуль транспортних кадрів), у виводі з'являється повідомлення *"Створено зображення без транспорту"*, а для подальшої обробки формується сірий заповнювач, який використовується як умовне зображення пустої дороги. Це є важливим елементом для тестування стабільності системи, яка повинна коректно реагувати на відсутність вхідних даних.

```

--- Напрямок 4 ---
Вибрано 14 зображень.
C:\Users\tshev/.cache\torch\hub\ultralytics_yolov5_master\models\common.py:906: FutureWarning: `torch.cuda.am
'
with amp.autocast(autocast):
Збережено об'єднане зображення: C:/Users/tshev/Desktop/Processed_Images\Dir_4\combined_1746542086.jpg
Загальна кількість ТЗ: 15
Повільний транспорт: 5
Час зеленої фази: 50 секунд.
```

Рисунок 3.9-Вивід результатів імітаційного моделювання

Рисунок 3.9 демонструє вивід результатів імітаційного моделювання транспортного потоку для одного з напрямків. У разі наявності зображень, система проводить їх комбінування в одне загальне зображення та передає на обробку детектору об'єктів. Після аналізу в консолі відображається шлях до збереженого об'єданого зображення з анотаціями: *"Збережено об'єднане зображення: ..."*. Це дозволяє розробнику швидко перейти до перевірки візуального результату детекції.

Найважливіша частина аналізу — це облік кількості виявлених транспортних засобів та розподіл їх за типами. Вивід *"Загальна кількість ТЗ: ..."* дає загальну оцінку трафіку на напрямку, тоді як *"Повільний транспорт: ..."* вказує кількість важкого транспорту (вантажівки, автобуси), що мають більший пріоритет при прийнятті рішень. Це співвідношення між легким і важким транспортом лягає в основу алгоритму обчислення вагової інтенсивності руху.

Також система обчислює тривалість зеленої фази відповідно до кількості транспортних засобів і їх ваги. Результат виводиться у форматі *"Час зеленої фази: ... секунд."* Ця інформація дозволяє оцінити ефективність функції визначення часу зеленого сигналу яка, враховуючи вагові коефіцієнти для легкого та важкого транспорту, встановлює адаптивну тривалість фази у межах допустимого діапазону (від 15 до 65 секунд). Якщо вагова інтенсивність перевищує заданий поріг, виводиться максимальне значення, що свідчить про перевантажений напрямок.

```

--- Напрямок 1 ---
Вибрано 0 зображень.
Створено зображення без транспорту.
Помилка при обробці зображень.
Загальна кількість ТЗ: 0
Повільний транспорт: 0
Пропускаємо напрямок.

```

Рисунок 3.10-Вивід результатів імітаційного моделювання при відсутності транспорту

Якщо кількість виявлених ТЗ дорівнює нулю, у виводі з'являється повідомлення *"Пропускаємо напрямок"*, як на рисунку 3.10, після чого система переходить до аналізу наступного напрямку без активації зеленої фази. Така поведінка підтверджує адаптивність алгоритму — зелене світло не вмикається без потреби, що знижує загальні затримки на перехресті.

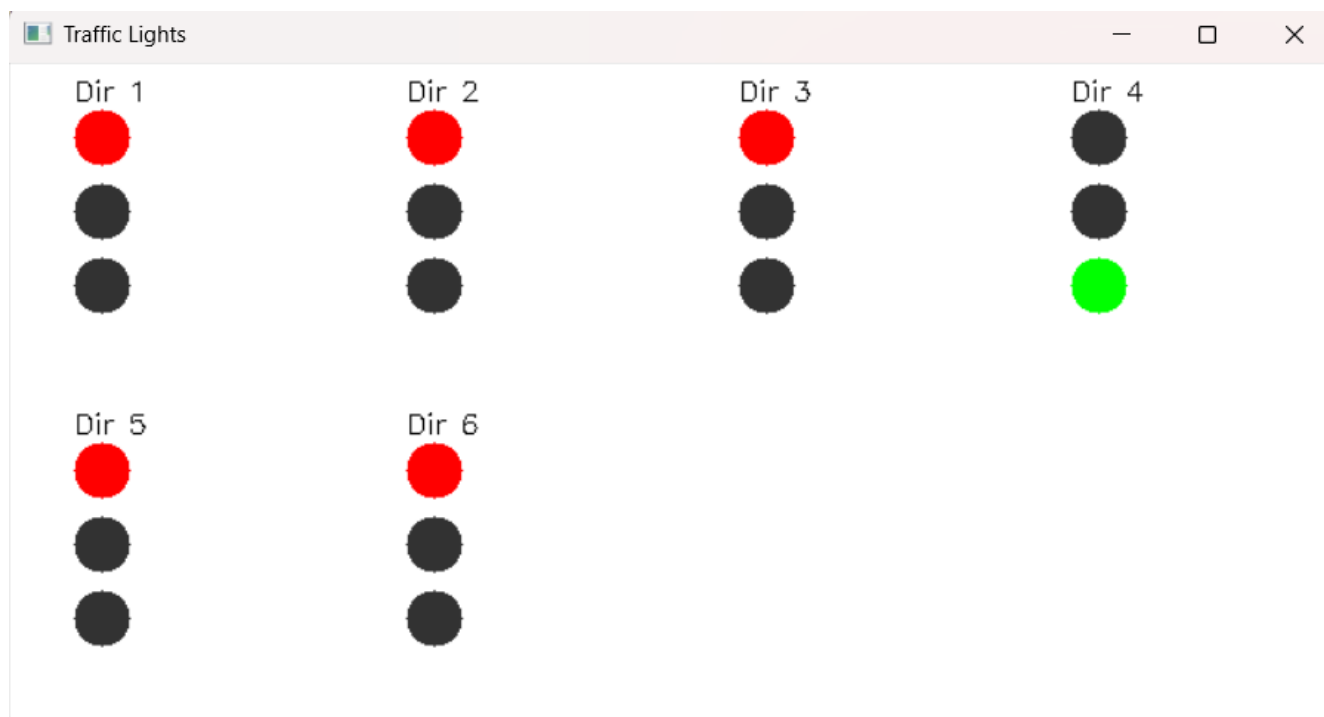


Рисунок 3.11-Візуалізація світлофорних фаз

Крім того, моделювання передбачає візуалізацію роботи світлофорів. Виведення світлофорних фаз супроводжується графічним оновленням вікна, де для кожного напрямку змінюється стан сигнальних ламп, як на рисунку 3.11. Це дозволяє у режимі реального часу спостерігати за роботою алгоритму, забезпечуючи наочність результатів моделювання.

Загалом, аналіз інформації з виводів в консоль дозволив не лише підтвердити правильність логіки роботи системи, а й виявити діапазони реакцій на різні ситуації: від повної відсутності транспорту до високої інтенсивності, що імітує годину пік. Такі умови є критичними для перевірки адаптивності системи, її стабільності при змінному навантаженні, та загальної коректності роботи в межах заданих алгоритмів.

3.3 Порівняльний аналіз ефективності програмного рішення з аналогами

Розроблене програмне забезпечення є втіленням синтезу різноманітних методик для створення дієвої імітаційної моделі адаптивного фазового управління

світлофорами. Модель ґрунтується на всебічному аналізі інтенсивності транспортного потоку, що здійснюється шляхом опрацювання даних з відеокамер та індукційних петель, що дає змогу отримати багатовимірне розуміння дорожньої обстановки.

Для об'єктивної оцінки функціональності запропонованого рішення було виконано ретельне порівняння з наявними аналогами. Цей аналіз надає можливість чітко визначити сильні та слабкі сторони кожного підходу, а також з'ясувати, за яких обставин розроблене рішення є найефективнішим. Цінність подібного зіставлення полягає не тільки у демонстрації інноваційності розробки, а й у обґрунтуванні її доцільності порівняно зі стандартними та комерційними системами.

Таблиця 3.1

Порівняльна характеристика типів систем управління дорожнім рухом

Характеристика	Розроблене рішення	Класичні фіксовані таймери	Комерційні адаптивні системи (SCOOT/SCATS)
Тип детекції транспорту	Камери + індукційні петлі	Відсутня або петлі	Детектори, камери
Модель виявлення об'єктів	YOLOv5	Немає	Пропріетарні алгоритми
Облік типу ТЗ	Так (легкі/важкі)	Ні	Частково
Адаптація тривалості фаз	Так (динамічно)	Ні (статична)	Так (динамічно, з прогнозуванням)
Гнучкість кількості напрямків в	Висока	Середня	Обмежена

Продовження таблиці 3.1

Можливість локальної інсталяції	Так (Raspberry Pi)	Так	Ні (централізована інфраструктура)
Простота розгортання та налаштування	Висока (відкрите ПЗ)	Висока	Низька (висока вартість і складність)
Вартість впровадження	Низька	Низька	Висока

Як чітко видно з вищевказаної таблиці, запропоноване рішення являє собою збалансований підхід, що враховує ключові аспекти: ціну, точність та адаптивність. Це свідчить про те, що система не лише ефективно регулює рух, але й є економічно вигідною та легко впроваджується. Використання нейронної мережі є ключовою особливістю розробленого рішення. Ця модель дозволяє системі не просто визначати кількість транспортних засобів, що є стандартною функцією для багатьох систем, а й класифікувати їх за типом, розрізняючи, наприклад, легкові авто, вантажівки та автобуси. Така деталізована інформація сприяє більш ефективному керуванню тривалістю зеленого сигналу, оскільки система може динамічно коригувати час, враховуючи інтенсивність руху різних типів транспорту. Наприклад, якщо на перехресті переважають вантажні автомобілі, система може збільшити тривалість зеленої фази для забезпечення їх плавного проїзду.

У порівнянні з традиційними системами управління дорожнім рухом, які застосовують фіксовані таймери, перевага розробленої системи в адаптивності очевидна. Звичайні системи, як правило, працюють за заздалегідь визначеними циклами, які не змінюються в залежності від поточної дорожньої ситуації. Це може призводити до значних затримок та неефективного використання пропускної здатності доріг, особливо в умовах динамічного руху, де інтенсивність транспорту може суттєво змінюватися протягом дня. На відміну від цього, розроблена система

здатна динамічно регулювати тривалість зеленого сигналу на основі даних, отриманих у реальному часі, що дає змогу значно зменшити затори та оптимізувати потік транспорту. Крім того, розроблена система має високу гнучкість у масштабуванні для обслуговування будь-якої кількості напрямків, що є важливим фактором у плануванні та розвитку міської інфраструктури.

У порівнянні з комерційними адаптивними системами, такими як SCATS або SCOOT, розроблене рішення також має низку значних переваг. Хоча комерційні системи також здатні адаптувати тривалість фаз світлофорів, вони часто мають закриту архітектуру, що ускладнює їх інтеграцію з іншими системами та модифікацію під конкретні потреби. Крім того, вони зазвичай вимагають значних інвестицій в дорожу інфраструктуру, що робить їх менш доступними для багатьох міст. Натомість, розроблена система має відкриту архітектуру, яка забезпечує більшу гнучкість та адаптивність. Її також можна легко реалізувати на малопотужному обладнанні, що значно знижує вартість впровадження. Це робить її особливо привабливою для міст з обмеженим бюджетом, які прагнуть модернізувати свої системи управління дорожнім рухом. Більше того, розроблена система надає більше гнучкості в налаштуванні логіки прийняття рішень, що дозволяє адаптувати її до конкретних вимог кожного перехрестя або ділянки дороги.

Однак, необхідно визнати, що розроблена система, на відміну від промислових аналогів, таких як SCATS або SCOOT, які враховують майбутні транспортні потоки, наразі не використовує історичні дані та прогнозування. Це означає, що система реагує на поточну дорожню ситуацію, але не передбачає зміни дорожнього трафіку в найближчому майбутньому. Хоча це не є критичним недоліком, оскільки система все одно забезпечує суттєве покращення порівняно з традиційними системами, інтеграція можливостей прогнозування може стати важливим напрямком подальшого розвитку та підвищення ефективності системи в довгостроковій перспективі[60].

3.4 Валідація результатів та аналіз похибок

У процесі тестування розробленої системи керування транспортним рухом було проведено ретельну валідацію отриманих результатів та детальний аналіз виявлених похибок. Це є критично важливим етапом для оцінки надійності та практичної застосовності системи в реальних умовах експлуатації.

Детальне вивчення роботи розробленої системи виявило кілька типових видів збоїв, які впливають на її загальну коректність та надійність. Зокрема, значну увагу привертають помилкові спрацьовування індукційних петель, що є важливим компонентом системи розпізнавання транспортних засобів. Індукційна петля – це, по суті, дріт, вмонтований у дорожнє покриття, функція якого – визначення проїзду транспортного засобу через зміну індуктивності при його знаходженні в радіусі дії петлі. Проте, в процесі експлуатації системи можуть виникати випадки, коли петля сигналізує про наявність транспорту, навіть коли над нею фактично немає жодного об'єкта.

Такі помилкові спрацьовування можуть бути зумовлені різними факторами, кожен з яких вносить вклад у загальну картину похибок. Однією з можливих причин є вплив електромагнітних перешкод від інших електричних пристроїв або ліній електропередач, розміщених поблизу місця встановлення петлі. Ці перешкоди можуть індукувати електричний струм в ланцюгу петлі, імітуючи присутність транспортного засобу та приводячи до неправильної реєстрації. Окрім того, короточасні перебої в електропостачанні, навіть незначні коливання напруги, можуть спричинити нестабільну роботу контролера петлі та спровокувати помилкові сигнали. Не слід виключати вплив погодних умов, особливо таких явищ, як сильний дощ або різкі перепади температури. Вода, яка накопичується в тріщинах дорожнього полотна, може змінити електричні характеристики петлі, а коливання температури – вплинути на параметри електронних компонентів контролера.

Щоб зменшити негативний вплив цих помилкових спрацьовувань на загальну точність системи, певні механізми обробки сигналів вже застосовані на

рівні контролерів індукційних контурів. Багато сучасних контролерів оснащені вбудованими алгоритмами фільтрації, які покликані пом'якшити вплив шумів та перешкод.

Існують різноманітні алгоритми, які можливо застосувати для реалізації таких фільтрів. Наприклад, одним з простих, але ефективних підходів є застосування часових фільтрів, котрі ігнорують сигнали, що є коротшими за певний поріг. Оскільки для проїзду реального автомобіля потрібен певний час, короткочасні сплески сигналу можна трактувати як аномалії. Інший підхід полягає у використанні статистичних фільтрів, що аналізують статистичні характеристики сигналу, як-от середнє значення та дисперсія, і відкидають значення, котрі значно відхиляються від очікуваних. Більш складні алгоритми можуть використовувати методи машинного навчання для класифікації сигналів та виявлення складних закономірностей, що відрізняють реальні проїзди від помилкових спрацьовувань. Варто підкреслити, що вибір конкретного алгоритму фільтрації, якщо він використовується, залежить від характеристик індукційних петель, що застосовуються, та рівня перешкод на конкретному місці установки.

Іншим важливим типом помилок, які здатні виникнути в розробленій системі, є невірне розпізнавання об'єктів алгоритмом комп'ютерного зору. Точність виявлення автотранспорту за допомогою YOLOv5 може бути дуже чутливою до певних ускладнень, що часто спостерігаються в реальних дорожніх умовах.

Зокрема, труднощі з розпізнаванням об'єктів можуть бути спровоковані наявністю тіней на дорожньому полотні. Тіні, які відкидають різні об'єкти, можуть частково маскувати транспортні засоби, потенційно утруднюючи їх розпізнавання алгоритмом. У таких ситуаціях система може мати труднощі з чітким визначенням контурів машин, що теоретично здатне призвести до помилок у підрахунку їх кількості або навіть до пропуску деяких об'єктів.

Ще однією умовою, яка може негативно вплинути на точність, є висока інтенсивність руху. За умови великої кількості машин, які рухаються щільно одна до одної, існує ймовірність часткового перекриття транспортних засобів. Це може ускладнити алгоритму відокремлення окремих автомобілів, що потенційно може

призвести до помилкової ідентифікації групи машин як єдиного об'єкта або, навпаки, до пропуску автомобілів, частково перекритих іншими.

До того ж, на точність виявлення може вплинути складне тло. Ділянки дороги поблизу рекламних щитів, споруд з різною архітектурою чи інших об'єктів можуть створювати візуальний шум, що ускладнює алгоритму виокремлення транспортних засобів з навколишнього середовища.

Щоб кількісно оцінити точність системи, було співставлено фактичну та виявлену кількість автомобілів у різних змодельованих умовах.

Таблиця 3.2

Точність виявлення автомобілів у різних сценаріях руху

Сценарій моделювання	Фактична кількість автомобілів	Виявлена кількість автомобілів	Відсоток похибки
Стабільний рух	12	12	0%
Збільшення інтенсивності	23	21	91.3%
Короткочасний затор	28	26	92.85%

Аналізуючи дані таблиці 3.1, спостерігаємо різну точність виявлення транспортних засобів системою в залежності від імітованих сценаріїв руху.

Сценарій "Сталий рух", що відображає ідеалізовану ситуацію з рівномірним розподілом машин та відсутністю перешкод, продемонстрував високий рівень точності. Під час моделювання потоку з 12 автомобілями, система бездоганно виявила всі 12 транспортних засобів. Це підтверджує її здатність ефективно функціонувати в умовах низької інтенсивності руху та відсутності перешкод. Варто

відзначити, що система не припустилася жодної помилки в цьому сценарії, що є зразком точності.

Переходячи до сценарію "Зростання інтенсивності", який імітує збільшення кількості транспортних засобів на дорозі, типово для годин пік або ділянок з інтенсивним трафіком, кількість змодельованих автомобілів була збільшена до 23. Система виявила 21 автомобіль, що вказує на невелике зниження точності порівняно зі сценарієм "Сталий рух". Похибка у цьому випадку склала 9,13%, що показує, що система все ще може досить точно розпізнавати більшість автомобілів, хоча і починає відчувати труднощі з збільшенням їх кількості.

Сценарій "Короткочасний затор" вважається найскладнішим з трьох. Він імітує ситуацію з утворенням затору на дорозі, характеризуючись високою щільністю транспортних засобів та їх близькістю. У цьому сценарії кількість змодельованих автомобілів була збільшена до 28. Система виявила 26 автомобілів, що вказує на подальше, хоча і незначне, зниження точності. Похибка у цьому випадку склала 7,15%. Важливо наголосити, що, незважаючи на підвищену складність сценарію, система все ще здатна виявляти значну кількість транспортних засобів з відносно невеликою похибкою.

Підсумовуючи результати моделювання, можна зробити висновок, що розроблена система демонструє високу точність виявлення автомобілів, особливо в стабільних умовах руху, коли щільність транспортних засобів низька, а рух рівномірний. Це свідчить про ефективність алгоритмів обробки зображень та їх здатність правильно ідентифікувати транспортні засоби за відсутності серйозних перешкод.

Проте, зі збільшенням інтенсивності руху та ускладненням сценаріїв моделювання, таких як затори чи інші фактори, що впливають на виявлення, точність системи дещо зменшується. Це може бути пов'язано з різними причинами, включаючи часткове перекриття транспортних засобів, їх близькість, зміни освітлення або інші фактори, що ускладнюють аналіз зображення.

Слід зазначити, що висновки, зроблені на основі даних у таблиці 3.1, базуються на аналізі випадкових імітаційних зображень, створених тестовою

системою. Ці колажі є спрощеною моделлю дорожнього руху. Однак, загальна тенденція показує, що система здатна досить точно виявляти автомобілі. У більшості випадків розбіжності між фактичною та виявленою кількістю незначні, і полягають у тому, що система пропускає лише невелику кількість транспортних засобів.

ВИСНОВКИ

У дипломній роботі вдало впроваджена інтелектуальна система управління трафіком, що є результатом поєднання передових методів комп'ютерного зору з традиційними технологіями збору даних від індукційних петель, встановлених на дорозі. Запропонований комплексний підхід забезпечує значно глибше розуміння поточної ситуації на дорогах у реальному часі. Це досягається за рахунок того, що система аналізує не лише присутність транспортних засобів в зоні контролю, але й їх точну кількість та тип.

Ретельний аналіз результатів наочно продемонстрував, що об'єднання даних з різних сенсорних джерел є ключем до значного підвищення ефективності адаптивного управління світлофорами на міських перехрестях. Такий комплексний підхід суттєво збільшує точність виявлення транспортних засобів, а також сприяє зменшенню потенційних помилкових спрацьовувань або ігнорування інтенсивності трафіку, особливо в складних погодних умовах, таких як сильний дощ чи сніг, а також при недостатньому освітленні в нічний час або в умовах туману.

Основою розробленої системи є одноплатний комп'ютер Raspberry Pi, який виконує роль центрального процесора, координуючи роботу всіх підсистем. Цей компактний комп'ютер підключений до декількох USB-камер високої роздільної здатності, що стратегічно розташовані вздовж основних напрямків руху на перехресті, а також до традиційних індукційних петель, які забезпечують стабільну роботу з великим об'ємом фізичних сигналів у режимі реального часу. Архітектура програмного забезпечення системи характеризується модульністю, що забезпечує значні переваги з точки зору гнучкості та зручності. Така структура дозволяє легко налаштовувати та оперативно оновлювати окремі функціональні компоненти без необхідності втручання в роботу всієї системи та без негативного впливу на її загальну продуктивність.

Ключовим елементом аналітичного блоку розробленої інтелектуальної системи є попередньо навчена модель глибокого навчання YOLOv5, яка відповідає

за точне виявлення різних типів транспортних засобів безпосередньо на отриманих зображеннях з камер. Результати цієї потужної моделі комп'ютерного зору далі використовуються спеціалізованим логічним модулем. На основі цих даних модуль виконує складні розрахунки для визначення оптимальної тривалості зеленої фази світлофора для кожного з напрямків руху, враховуючи поточний транспортний потік. Окремо в системі передбачено механізм врахування зниження видимості на дорозі. В таких умовах система автоматично переходить на пріоритетне використання даних, що надходять від надійних індукційних контурів, забезпечуючи стабільну роботу.

Результати комплексних випробувань розробленої системи в контрольованому середовищі імітації продемонстрували її стабільну та надійну роботу, високу здатність оперативно реагувати на динамічні зміни дорожньої ситуації в реальному часі, а також значний потенціал для подальшої повної автоматизації процесу управління трафіком на міських перехрестях. Завдяки обраній гнучкій та масштабованій архітектурі, система може бути легко адаптована для перехресть з будь-якою кількістю напрямків руху та конфігурацій, а також може бути ефективно пристосована до особливостей дорожньої інфраструктури конкретного міста. Важливим аспектом розробленого рішення є його відносно невисока вартість, що робить його реалістичною та привабливою альтернативою дорогим комерційним рішенням, які часто є недоступними для міст з обмеженим фінансовим ресурсом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Traffic flow theories. *Traffic theory*. Boston. P. 1–44.
URL: https://doi.org/10.1007/0-306-48217-7_1 (дата звернення: 13.05.2025).
2. Goyal P., Goyal S., Kaur N. Review various technologies used to manage Traffic Management System using IOT. *CGC international journal of contemporary technology and research*. 2021. Vol. 4, no. 1. P. 235–239.
URL: <https://doi.org/10.46860/cgcijctr.2021.12.31.235> (дата звернення: 13.05.2025).
3. Pomaska G. Stereo vision applying opencv and raspberry pi. *ISPRS - international archives of the photogrammetry, remote sensing and spatial information sciences*. 2019. XLII-2/W17. P. 265–269. URL: <https://doi.org/10.5194/isprs-archives-xlii-2-w17-265-2019> (дата звернення: 13.05.2025).
4. Cherrett T., Bell H., McDonald M. Traffic management parameters from single inductive loop detectors. *Transportation research record: journal of the transportation research board*. 2000. Vol. 1719, no. 1. P. 112–120.
URL: <https://doi.org/10.3141/1719-14> (дата звернення: 13.05.2025).
5. Conceptual approach to hybrid traffic management in cities / L. Abramova et al. *Lecture notes in networks and systems*. Cham, 2025. P. 81–89.
URL: https://doi.org/10.1007/978-3-031-87379-9_8 (дата звернення: 13.05.2025).
6. A fuzzy logic-based smart traffic management systems / A. A. Abdou et al. *Journal of computer science*. 2022. Vol. 18, no. 11. P. 1085–1099.
URL: <https://doi.org/10.3844/jcssp.2022.1085.1099> (дата звернення: 13.05.2025).
7. Smart traffic light management systems / A. A. A. R. Magableh et al. *International journal of technology diffusion*. 2020. Vol. 11, no. 3. P. 22–47.
URL: <https://doi.org/10.4018/ijtd.2020070102> (дата звернення: 13.05.2025).
8. Elefteriadou L. Simulation modeling. *An introduction to traffic flow theory*. Cham, 2024. P. 177–204. URL: https://doi.org/10.1007/978-3-031-54030-1_8 (дата звернення: 13.05.2025).
9. Mahesh Kumar G., Kumaraswamy E. Smart traffic junction using raspberry pi. *IOP conference series: materials science and engineering*. 2020. Vol. 981. P. 032048.
URL: <https://doi.org/10.1088/1757-899x/981/3/032048> (дата звернення: 13.05.2025).

10. Load estimation of moving passenger cars using inductive-loop technology / Z. Marszalek et al. *Sensors*. 2023. Vol. 23, no. 4. P. 2063. URL: <https://doi.org/10.3390/s23042063> (дата звернення: 13.05.2025).

11. Adaptive traffic light controller based on congestion detection using computer vision / R. G. Putra et al. *Journal of physics: conference series*. 2021. Vol. 1845, no. 1. P. 012047. URL: <https://doi.org/10.1088/1742-6596/1845/1/012047> (дата звернення: 13.05.2025).

12. Computer vision for vehicle detection: a comprehensive review / S. El Asri et al. *Data and metadata*. 2025. Vol. 4. P. 873. URL: <https://doi.org/10.56294/dm2025873> (дата звернення: 13.05.2025).

13. Manikkannan P. D. Vehicle tracking and counting through image processing. *Interantional journal of scientific research in engineering and management*. 2023. Vol. 07, no. 12. P. 1–11. URL: <https://doi.org/10.55041/ijrsrem27599> (дата звернення: 14.05.2025).

14. A simulation-based traffic signal control for congested urban traffic networks / S. Baldi et al. *Transportation science*. 2019. Vol. 53, no. 1. P. 6–20. URL: <https://doi.org/10.1287/trsc.2017.0754> (дата звернення: 14.05.2025).

15. Realtime data traffic analyser locomotive of big data analytics. *International journal of recent technology and engineering*. 2020. Vol. 8, no. 6. P. 5643–5646. URL: <https://doi.org/10.35940/ijrte.f9809.038620> (дата звернення: 13.05.2025).

16. IOT based traffic management system / S. S et al. *International journal of scientific research in computer science, engineering and information technology*. 2021. P. 184–187. URL: <https://doi.org/10.32628/cseit1217451> (дата звернення: 13.05.2025).

17. Machine learning in ITS / W. A. Altabey et al. *Modeling and simulation of intelligent transportation systems*. Boca Raton, 2024. P. 80–95. URL: <https://doi.org/10.1201/9781032691787-5> (дата звернення: 13.05.2025).

18. Deshpande S., Muron W., Cai Y. Vehicle classification. *Computer vision and imaging in intelligent transportation systems*. Chichester, UK, 2017. P. 47–79. URL: <https://doi.org/10.1002/9781118971666.ch3> (дата звернення: 13.05.2025).

19. A comprehensive review of vehicle detection using computer vision / A. F. Abbas et al. *TELKOMNIKA (telecommunication computing electronics and control)*. 2021. Vol. 19, no. 3. P. 838. URL: <https://doi.org/10.12928/telkomnika.v19i3.12880> (дата звернення: 13.05.2025).

20. Marmayohan N., Farah A. Scene analysis using Tensorflow & YOLO algorithms on Raspberry pi 4 : thesis. 2021. URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-45540> (дата звернення: 13.05.2025).

21. GitHub - ultralytics/yolov5: YOLOv5 in PyTorch > ONNX > CoreML > TFLite. *GitHub*. URL: <https://github.com/ultralytics/yolov5> (дата звернення: 13.05.2025).

22. Truong S. N. A low-cost artificial neural network model for raspberry pi. *Engineering, technology & applied science research*. 2020. Vol. 10, no. 2. P. 5466–5469. URL: <https://doi.org/10.48084/etasr.3357> (дата звернення: 14.05.2025).

23. Cohen J. Faster RCNN in 2025: How it works and why it's still the benchmark for Object Detection. *Welcome to The Library!*. URL: <https://www.thinkautonomous.ai/blog/faster-rcnn/> (дата звернення: 13.05.2025).

24. Singh R. Understanding and implementing faster R-CNN. *Medium*. URL: <https://medium.com/@RobuRishabh/understanding-and-implementing-faster-r-cnn-248f7b25ff96> (дата звернення: 13.05.2025).

25. Hui J. SSD object detection: Single Shot MultiBox Detector for real-time processing. *Medium*. URL: <https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06> (дата звернення: 13.05.2025).

26. Object detection: yolo vs faster r-cnn. *International research journal of modernization in engineering technology and science*. 2022. URL: <https://doi.org/10.56726/irjmets30226> (дата звернення: 13.05.2025).

27. Computer vision based research on PCB recognition using SSD neural network / D. Li et al. *Journal of physics: conference series*. 2021. Vol. 1815, no. 1.

P. 012005. URL: <https://doi.org/10.1088/1742-6596/1815/1/012005> (дата звернення: 13.05.2025).

28. A multiple inductive loop vehicle detection system for heterogeneous and lane-less traffic / S. Sheik Mohammed Ali et al. *IEEE transactions on instrumentation and measurement*. 2012. Vol. 61, no. 5. P. 1353–1360. URL: <https://doi.org/10.1109/tim.2011.2175037> (дата звернення: 13.05.2025).

29. Boston Traffic and Parking Dept, Corporation T. R., Authority B. R. Coordinated traffic signal system for downtown boston. Creative Media Partners, LLC, 2018.

30. Kustija J. SCATS (sydney coordinated adaptive traffic system) as A solution to overcome traffic congestion in big cities. *International journal of research and applied technology*. 2023. Vol. 3, no. 1. P. 1–14. URL: <https://doi.org/10.34010/injuratech.v3i1.7875> (дата звернення: 13.05.2025).

31. Sims A. G., Dobinson K. W. The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. *IEEE transactions on vehicular technology*. 1980. Vol. 29, no. 2. P. 130–137. URL: <https://doi.org/10.1109/t-vt.1980.23833> (дата звернення: 13.05.2025).

32. Stevanovic A., Martin P. T. Split-Cycle offset optimization technique and coordinated actuated traffic control evaluated through microsimulation. *Transportation research record: journal of the transportation research board*. 2008. Vol. 2080, no. 1. P. 48–56. URL: <https://doi.org/10.3141/2080-06> (дата звернення: 13.05.2025).

33. Smart urban signal networks: initial application of the SURTRAC adaptive traffic signal control system / S. Smith et al. *Proceedings of the international conference on automated planning and scheduling*. 2013. Vol. 23. P. 434–442. URL: <https://doi.org/10.1609/icaps.v23i1.13594> (дата звернення: 13.05.2025).

34. АСКДР - автоматизована система керування дорожнім рухом | компанія СЕА. Компанія СЕА | Виробництво електроніки, електронні компоненти, електротехніка | Компанія СЕА. URL: <https://www.sea.com.ua/ua/svetofofnaya-produkciya/kompleksnaa-sistema-upravlenia-doroznym->

[dvizeniem/?srsltid=AfmBOopuKkwgxHrOvkMgSY7o1VwAV_S2Y8cCQxBy2Te6Um-YCGa6hgBO](http://elartu.tntu.edu.ua/handle/lib/35565) (дата звернення: 13.05.2025).

35. Гузар Д. Р., Huzar D. Комп'ютеризована система розпізнавання образів на основі Raspberry PI : bachelor's thesis. 2021. URL: <http://elartu.tntu.edu.ua/handle/lib/35565> (дата звернення: 13.05.2025).

36. Raspberry pi. Movement Publishing, 2021.

37. Phanindra J. N. Raspberry pi based traffic density monitoring and controlling system. *International journal for research in applied science and engineering technology*. 2017. Vol. V, no. VIII. P. 393–400. URL: <https://doi.org/10.22214/ijraset.2017.8055> (дата звернення: 13.05.2025).

38. Welcome to python.org. *Python.org*. URL: <https://www.python.org/> (дата звернення: 13.05.2025).

39. Home. *OpenCV*. URL: <https://opencv.org> (дата звернення: 13.05.2025).

40. S P., S N., S K. USB device driver development of image processing using raspberry pi 4 via thingspeak. *International journal of electronics and communication engineering*. 2021. Vol. 8, no. 6. P. 1–3. URL: <https://doi.org/10.14445/23488549/ijece-v8i6p101> (дата звернення: 13.05.2025).

41. Kester W. Analog switches and multiplexers. *Data conversion handbook*. 2005. P. 493–517. URL: <https://doi.org/10.1016/b978-075067841-4/50030-0> (дата звернення: 13.05.2025).

42. Binary decoder used to decode a binary codes. *Basic Electronics Tutorials*. URL: https://www.electronics-tutorials.ws/combinational/comb_5.html (дата звернення: 13.05.2025).

43. Dixit Y., Mohanty G., Arockiasamy K. Adaptive traffic control. *IoT and analytics in renewable energy systems (volume 2)*. Boca Raton, 2023. P. 181–187. URL: <https://doi.org/10.1201/9781003374121-16> (дата звернення: 13.05.2025).

44. Abdul Kadhar K. M., Anand G. Sensors and signals. *Data science with raspberry pi*. Berkeley, CA, 2021. P. 79–97. URL: https://doi.org/10.1007/978-1-4842-6825-4_4 (дата звернення: 14.05.2025).

45. Rajeshwari R. B., Mahaveera K. Prototype for parking shield system using raspberry pi. *Advances in VLSI, signal processing, power electronics, iot, communication and embedded systems*. Singapore, 2023. P. 1–13. URL: https://doi.org/10.1007/978-981-99-4444-6_1 (дата звернення: 13.05.2025).
46. Smart street light system using raspberry pi / S. Kadam et al. *SSRN electronic journal*. 2021. URL: <https://doi.org/10.2139/ssrn.3871011> (дата звернення: 13.05.2025).
47. Subagja M., Rahman B. Object Detection Using YOLOv5 and OpenCV. *PIKSEL : penelitian ilmu komputer sistem embedded and logic*. 2025. Vol. 13, no. 1. P. 155–162. URL: <https://doi.org/10.33558/piksel.v13i1.10772> (дата звернення: 13.05.2025).
48. Aboyomi D. D., Daniel C. A Comparative Analysis of Modern Object Detection Algorithms: YOLO vs. SSD vs. Faster R-CNN. *ITEJ (information technology engineering journals)*. 2023. Vol. 8, no. 2. P. 96–106. URL: <https://doi.org/10.24235/itej.v8i2.123> (дата звернення: 13.05.2025).
49. Turull T. D. Open source traffic analyzer : thesis. 2010. URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-38322> (дата звернення: 13.05.2025).
50. Cicolani J. Raspberry Pi GPIO. *Beginning robotics with raspberry pi and arduino*. Berkeley, CA, 2018. P. 103–128. URL: https://doi.org/10.1007/978-1-4842-3462-4_4 (дата звернення: 13.05.2025).
51. Gay W. Uart. *Advanced raspberry pi*. Berkeley, CA, 2018. P. 167–187. URL: https://doi.org/10.1007/978-1-4842-3948-3_10 (дата звернення: 13.05.2025).
52. Gay W. Uart. *Advanced raspberry pi*. Berkeley, CA, 2018. P. 89–119. URL: https://doi.org/10.1007/978-1-4842-3948-3_10 (дата звернення: 13.05.2025).
53. Aiyappa R. Section 24. inter-integrated circuit (I2C). Microchip Technology Incorporated, 2016.
54. Sathish Rao. Traffic signal timing and real time optimization. *International research journal on advanced engineering hub (IRJAEH)*. 2025. Vol. 3, no. 05. P. 2082–2089. URL: <https://doi.org/10.47392/irjaeh.2025.0304> (дата звернення: 14.05.2025).

55. Gay W. SPI bus. *Advanced raspberry pi*. Berkeley, CA, 2018. P. 283–302.
URL: https://doi.org/10.1007/978-1-4842-3948-3_16 (дата звернення: 13.05.2025).
56. Gay W. SPI bus. *Advanced raspberry pi*. Berkeley, CA, 2018. P. 177–201.
URL: https://doi.org/10.1007/978-1-4842-3948-3_16 (дата звернення: 13.05.2025).
57. Gay W. Ethernet. *Advanced raspberry pi*. Berkeley, CA, 2018. P. 131–153.
URL: https://doi.org/10.1007/978-1-4842-3948-3_8 (дата звернення: 13.05.2025).
58. Gay W. Ethernet. *Advanced raspberry pi*. Berkeley, CA, 2018. P. 71–82.
URL: https://doi.org/10.1007/978-1-4842-3948-3_8 (дата звернення: 13.05.2025).
59. Ziemann V. Example: data acquisition system. *A hands-on course in sensors using the arduino and raspberry pi*. 2nd ed. Boca Raton, 2023. P. 219–228.
URL: <https://doi.org/10.1201/9781003341703-13> (дата звернення: 13.05.2025).
60. Govindaraju S. Intelligent transportation system's machine learning-based traffic prediction. *Journal of applied data sciences*. 2024. Vol. 5, no. 4. P. 1826–1837.
URL: <https://doi.org/10.47738/jads.v5i4.364> (дата звернення: 13.05.2025).

Демонстраційні матеріали (презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ КАФЕДРА ІНФОРМАЦІЙНИХ
СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

НА ТЕМУ:

«ІНТЕГРОВАНА СИСТЕМА КЕРУВАННЯ ТРАНСПОРТНИМ РУХОМ З ВИКОРИСТАННЯМ
ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ ТА ІНДУКЦІЙНОЇ ПЕТЛІ»

Виконав:

Шевчук Віктор Володимирович

Керівник:

PhD Данильченко Валентина Миколаївна



АКТУАЛЬНІСТЬ ТЕМИ

- Зростання трафіку в містах потребує адаптивних рішень.
- Традиційні системи неефективні в умовах змінного навантаження.
- Необхідне інтелектуальне управління з використанням сучасних технологій.

МЕТА, ОБ'ЄКТ, ПРЕДМЕТ ТА ЗАВДАННЯ ДОСЛІДЖЕННЯ

Мета дослідження: Розробка адаптивної системи керування дорожнім рухом на основі інтеграції комп'ютерного зору та індукційних петель.

Об'єкт дослідження: Інтелектуальні системи керування транспортним рухом.

Предмет дослідження: Методи аналізу транспортних потоків з використанням комп'ютерного зору та сенсорів індукційного типу.

Завдання дослідження:

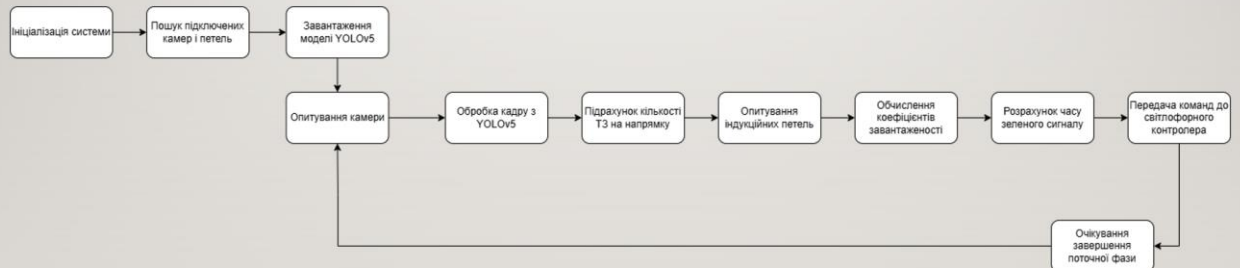
- Реалізувати робочу систему, що обробляє зображення з камер та дані з індукційної петлі.
- Провести тестування та оцінити ефективність запропонованого рішення.
- Порівняти з існуючими системами адаптивного керування транспортом.

ОСНОВНІ ТЕХНОЛОГІЇ

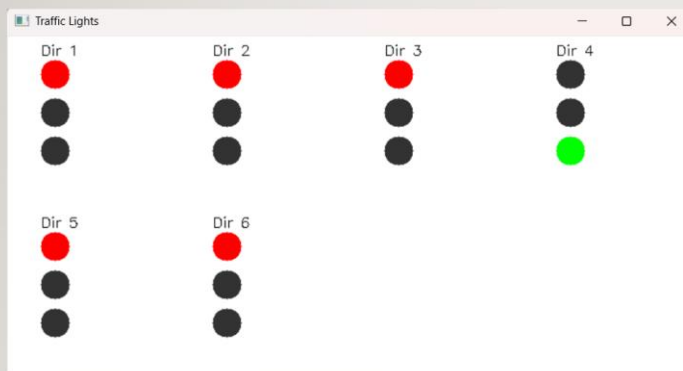
- Система поєднує камери, індукційні петлі, комп'ютерний зір (YOLOv5) та Raspberry Pi. Вибір компонентів обґрунтовано з урахуванням обмежених ресурсів.



АЛГОРИТМ РОБОТИ



МОДЕЛЮВАННЯ ТА СИМУЛЯЦІЯ



ПОРІВНЯННЯ З АНАЛОГАМИ

Характеристика	Розроблене рішення	Класичні фіксовані таймери	Комерційні адаптивні системи (SCOOT/SCATS)
Тип детекції транспорту	Камери + індукційні петлі	Відсутня або петлі	Детектори, камери
Модель виявлення об'єктів	YOLOv5	Немає	Пропріетарні алгоритми
Облік типу ТЗ	Так (легкі/важкі)	Ні	Частково
Адаптація тривалості фаз	Так (динамічно)	Ні (статична)	Так (динамічно, з прогнозуванням)
Гнучкість в кількості напрямків	Висока	Середня	Обмежена
Можливість локальної інсталяції	Так (Raspberry Pi)	Так	Ні (централізована інфраструктура)
Простота розгортання та налаштування	Висока (відкрите ПЗ)	Висока	Низька (висока вартість і складність)
Вартість впровадження	Низька	Низька	Висока

ПЕРСПЕКТИВИ РОЗВИТКУ

- Можливе розширення до потокового відео, прогнозування трафіку.
- Інтеграція з розумним містом, мобільними застосунками та паркуванням.

ВИСНОВКИ

- Проведено огляд сучасних систем адаптивного керування транспортом та виявлено їхні недоліки у вартості й складності впровадження.
- Розроблено систему з обробкою даних із камер і петель, реалізовано адаптивну логіку регулювання.
- Проведено тестування, яке підтвердило ефективність підходу в умовах обмежених ресурсів.

АПРОБАЦІЯ

- Назва тези «Підвищення ефективності та безпеки дорожнього руху шляхом інтеграції ІОТ» подана на VI Науково-технічна конференція «Сучасний стан та перспективи розвитку ІоТ»

ДЯКУЮ ЗА УВАГУ

