Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»

АВТОМАТИЗОВАНА СИСТЕМА МОНІТОРИНГУ ЗАВАНТАЖЕНОСТІ АВТОМОБІЛЬНИХ ДОРІГ

AUTOMATED ROAD CONGESTION MONITORING SYSTEM

спеціальність 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

Виконав: здобувач вищої освіти
групи КНм-21
Шемчук Петро Володимирович
_____
(підпис)
Керівник: д.пед.н., професор
Тулашвілі Юрій Йосипович
_____
(підпис)

Кваліфікаційну роботу
допущено до захисту
«___» _____ 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Валерій Олександрович
_____
(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерних наук*
Ступінь вищої освіти: *магістр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *122 Комп'ютерні науки*
Освітня програма: *«Комп'ютерні науки»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____Валерій ЛІЩИНА
«14» травня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА
ДРУГОГО (МАГІСТЕРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ

Шемчук Петро Володимирович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Автоматизована система моніторингу завантаженості автомобільних доріг»
Керівник д.пед.н., професор Тулашвілі Юрій Йосипович
затверджені наказом закладу вищої освіти від «14» травня 2025 р. № 255/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *«05» грудня 2025 р.*
3. Вихідні дані до роботи: _____
_____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів проектування, опис функціонального наповнення об'єкта проектування, розробка й обґрунтування системного наповнення, експерементальне дослідження результативності предмету дослідження.
_____
5. Перелік графічного матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз проблематики за темою роботи та постановка завдань дослідження	*Тулашвілі Ю. Й.*		
Теоретичне дослідження та практична реалізація предмету дослідження	*Тулашвілі Ю. Й.*		
Експеременталне дослідження результативності предмету дослідження	*Тулашвілі Ю. Й.*		
Показник запозичень тексту	_____ %		
Інструментальна перевірка	*Кошелюк В. А.*		
Нормоконтроль	*Сачук В. О.*		
Гарант ОПП	*Ліщина В. О.*		

7. Дата видачі завдання *«14» травня 2025 р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	*Провести огляд літературних джерел по темі кваліфікаційної роботи*	*до 30.06.2025 р*	
2	*Провести аналіз загальної проблеми і вибір напрямків дослідження*	*до 01.09.2025 р.*	
3	*Розробити функціональну схему роботи програмного продукту*	*до 01.10.2025 р*	
4	*Описати засоби розробки об'єкта проектування*	*до 15.10.2025 р.*	
5	*Практична реалізація об'єкта проектування*	*до 10.11.2025 р.*	
6	*Провести експеременталне дослідження результативності предмету дослідження*	*до 25.11.2025 р.*	
7	*Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру*	*до 05.12.2025 р.*	

Здобувач вищої освіти _____ Петро ШЕМЧУК

Керівник роботи _____ Юрій ТУЛАШВІЛІ

АНОТАЦІЯ

Шемчук П. В. Автоматизована система моніторингу завантаженості автомобільних доріг. Рукопис.

Кваліфікаційна робота магістра ОП «Комп'ютерні науки». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота магістра складається з вступу, 3 розділів, висновків, списку використаних джерел та додатків.

Кваліфікаційна робота присвячена розробленню та дослідженню автоматизованої системи моніторингу завантаженості автомобільних доріг на основі відеоданих і веборієнтованих інформаційних технологій. Метою роботи є створення прототипу системи, який реалізує повний цикл опрацювання даних: від отримання відеопотоків з камер спостереження до обчислення макроскопічних показників транспортного потоку, формування інтегрального індексу завантаженості та візуалізації результатів у вебінтерфейсі. Запропоновано багатошарову архітектуру, що включає модуль відеоаналітики, серверну частину з REST-інтерфейсом, базу даних та клієнтський вебзастосунок.

Розроблено методику експериментального дослідження з використанням репрезентативних відеофрагментів, ручної розмітки еталонних даних та метрик якості детекції (precision, recall, F1-мера) і похибок розрахунку інтегрального індексу (MAPE, RMSE). Експерименти показали прийнятну точність оцінювання завантаженості в денних умовах зйомки та невелику затримку оновлення даних у вебінтерфейсі, що підтверджує придатність прототипу для оперативного моніторингу обмеженої кількості ділянок дорожньої мережі.

Ключові слова: автоматизована система моніторингу, завантаженість автомобільних доріг, транспортний потік, інтегральний індекс завантаженості, комп'ютерний зір, відеоаналітика, вебзастосунок, інтелектуальні транспортні системи.

ABSTRACT

Petro Shemchuk. Automated road congestion monitoring system. Manuscript.

Master's thesis in Computer Science. Lutsk National Technical University. Lutsk, 2025.

The master's thesis consists of an introduction, 3 chapters, conclusions, a list of references, and appendices.

The master’s thesis focuses on the development and study of an automated road traffic congestion monitoring system based on video data and web-oriented information technologies. The aim of the research is to design a prototype that implements a full processing pipeline, from acquiring video streams from surveillance cameras to calculating macroscopic traffic flow parameters, computing an integral congestion index and visualizing the results in a web interface. A multilayer architecture is proposed, comprising a video analytics module, a server side with a REST API, a PostgreSQL database and a React-based client web application; UML component, class, sequence and deployment diagrams and a logical data model are provided together with key code fragments.

An experimental methodology is developed using representative video fragments from different road types, manual ground truth annotation and standard detection quality metrics (precision, recall, F1-score) as well as congestion index error measures (MAPE, RMSE). The experiments demonstrate acceptable congestion estimation accuracy in daylight conditions and a low delay of data updates in the web interface, which confirms that the prototype is suitable for near real-time monitoring of a limited set of road segments.

Keywords: automated monitoring system, road traffic congestion, traffic flow, integral congestion index, computer vision, video analytics, web application, intelligent transportation systems.

ЗМІСТ

ВСТУП

Збільшення кількості автомобільного транспорту, розширення приміських агломерацій та зростання мобільності населення призводять до постійного перевантаження дорожньої інфраструктури. Для значної частини міських жителів регулярні затори стали буденною реальністю, що зумовлює втрату часу, підвищення витрат пального, збільшення викидів шкідливих речовин і погіршення якості життя. Сучасні концепції «розумних міст» і стійкої урбаністики розглядають ефективне управління дорожнім рухом як одну з ключових умов сталого розвитку. У фокусі опиняються інтелектуальні транспортні системи, які спираються на постійний моніторинг стану дорожньої мережі та оперативний аналіз отриманих даних [1].

У науковій літературі й практичних впровадженнях виділяють кілька основних підходів до моніторингу завантаженості автомобільних доріг. До них належать інфраструктурні рішення на базі індукційних петель, радарів та інших стаціонарних сенсорів, краудсорсингові системи, які використовують дані від мобільних пристроїв, а також методи, засновані на комп’ютерному зорі та глибокому навчанні.

Огляд робіт з комп’ютерного зору для транспортного моніторингу показує, що сучасні детектори об’єктів на базі глибоких згорткових нейронних мереж забезпечують високу точність виявлення транспортних засобів і можуть працювати в режимі, наближеному до реального часу, навіть в умовах змінного освітлення і складних сцен [2].

Паралельно розвиваються IoT-орієнтовані системи моніторингу трафіку. У таких рішеннях дорожні сенсори, світлофори, камери та хмарні сервіси інтегруються в єдину платформу, що дає змогу збирати дані в реальному часі, зберігати часові ряди, аналізувати їх і надавати інформацію органам міського управління та громадянам.

Разом із тим для невеликих міст, закритих кампусів, індустріальних парків, територій університетів та логістичних центрів використання

комерційних глобальних сервісів не завжди є можливим або економічно доцільним. У таких випадках виникає потреба в гнучких і відкритих рішеннях, які можна реалізувати на базі доступної інфраструктури відеоспостереження та сучасних бібліотек комп'ютерного зору. Саме на розроблення такого рішення спрямована робота.

Об'єктом дослідження є процес моніторингу завантаженості автомобільних доріг у транспортній мережі. Предметом дослідження є моделі, методи і програмні засоби автоматизованого збирання, оброблення та візуалізації даних про завантаженість автомобільних доріг на основі відеопотоків і веб-технологій.

Метою роботи є розроблення та дослідження автоматизованої інформаційної системи моніторингу завантаженості автомобільних доріг, яка забезпечує отримання, оброблення й аналіз даних у режимі, наближеному до реального часу, з використанням методів комп'ютерного зору й сучасних вебтехнологій.

Для досягнення мети розв'язуються завдання аналізу предметної області та сучасних підходів до моніторингу трафіку, формування вимог до системи, розроблення архітектури та структур даних, реалізації прототипу системи на базі Python, Django REST Framework, PostgreSQL, OpenCV та React, а також експериментального дослідження точності й продуктивності розробленого рішення.

Наукова новизна полягає у формуванні інтегрованої моделі індексу завантаженості, адаптованої до використання відеопотоків з камер спостереження, а також у розробленні архітектури автоматизованої системи, що поєднує модуль відеоаналітики, REST-сервіси збору показників та вебінтерфейс для диспетчерів і аналітиків. Практичне значення полягає в можливості застосування прототипу як основи для локальних систем моніторингу дорожнього руху у містах і кампусах, а також як навчального стенду в освітньому процесі.

РОЗДІЛ 1
АНАЛІЗ ПРОБЛЕМАТИКИ АВТОМАТИЗОВАНОГО МОНІТОРИНГУ ЗАВАНТАЖЕНОСТІ АВТОМОБІЛЬНИХ ДОРІГ

1.1 Аналіз предметної області та показників завантаженості дорожньої мережі

У сучасних умовах інтенсивної урбанізації та зростання рівня автомобілізації питання оцінювання завантаженості дорожньої мережі набуває особливої актуальності. Для міських агломерацій характерними є регулярні затори, нерівномірність розподілу транспортних потоків у просторі та часі, значні втрати часу користувачів дорожньої інфраструктури й підвищення екологічного навантаження на навколишнє середовище. Ефективне управління дорожнім рухом неможливе без кількісної оцінки стану транспортного потоку на ключових ділянках мережі, що зумовлює необхідність формального опису предметної області та вибору адекватних показників завантаженості.

Дорожню мережу доцільно розглядати як множину взаємопов’язаних елементів: вулиць, проспектів, магістралей, перехресть, транспортних розв’язок та примикань. Для цілей моніторингу кожна ділянка моделюється як відрізок дороги з відносно однорідними геометричними та організаційними характеристиками, де фіксується транспортний потік, що складається із сукупності транспортних засобів, які рухаються в одному або кількох напрямках. Стан такого потоку описується набором макроскопічних показників, які узагальнюють характеристики руху на рівні групи транспортних засобів, а не окремих автомобілів.

Базовими макроскопічними показниками є інтенсивність руху, щільність потоку та середня швидкість. Інтенсивність, як правило, позначається символом q та характеризує кількість транспортних засобів, що перетинають контрольний поперечний перетин дороги за фіксований інтервал часу. У системі СІ інтенсивність вимірюють у транспортних засобах за годину або за секунду. Щільність потоку k описує кількість транспортних засобів, які

одночасно перебувають на одиниці довжини дороги, та вимірюється у транспортних засобах на кілометр. Середня швидкість v визначається як середнє значення швидкостей транспортних засобів, що рухаються по досліджуваній ділянці за обраний період часу.

Між цими величинами у стаціонарному режимі виконується відоме співвідношення $q = k \cdot v$, яке відображає, що інтенсивність є добутком щільності на середню швидкість. На основі цього співвідношення будується фундаментальна діаграма дорожнього руху, де інтенсивність q розглядається як функція від щільності k. На малих значеннях щільності збільшення кількості транспортних засобів призводить до зростання інтенсивності, оскільки зростає потік при збереженні відносно високої швидкості. Після досягнення певного критичного значення щільності подальше її збільшення спричиняє зниження середньої швидкості, що у свою чергу призводить до падіння інтенсивності та переходу потоку у режим нестійкого руху або затору. Такий характер залежності є фундаментальним для інтерпретації показників, які повинна вимірювати автоматизована система моніторингу [1-2].

У транспортній інженерії також широко використовується поняття рівня обслуговування, який характеризує суб’єктивне сприйняття умов руху водіями та пасажирами. Виділяють кілька рівнів обслуговування, від вільного руху до стану затору, що базуються на поєднанні таких кількісних характеристик, як середня швидкість, щільність, величина затримок та стабільність руху. Для автоматизованих систем моніторингу доцільно не лише фіксувати первинні величини q, k, v, а й формувати інтегральні індекси, які дозволяють у компактній формі відображати загальний стан потоку й спрощують інтерпретацію даних для диспетчерів та аналітиків [3].

У таблиці 1.1 узагальнено основні показники стану дорожнього руху, що використовуються в подальших розділах роботи. Для кожного з них наведено позначення, одиниці вимірювання, формальне визначення й приклади оцінювання в межах програмної системи, орієнтованої на аналіз відеопотоків з камер спостереження.

Таблиця 1.1 – Основні показники стану дорожнього руху

Показник	Позначення	Одиниця вимірювання	Формальне визначення	Приклад оцінювання у системі
Інтенсивність руху	q	транспортних засобів/год, транспортних засобів/с	Кількість транспортних засобів, що перетинають заданий поперечний перетин дороги за одиницю часу.	Підрахунок числа детектованих транспортних засобів, які перетнули віртуальну лінію підрахунку за інтервал 30-60 с.
Щільність транспортного потоку	k	транспортних засобів/км	Кількість транспортних засобів, які одночасно перебувають на одиниці довжини дорожньої ділянки.	Оцінювання кількості детектованих об’єктів у видимій зоні камери, нормованої на довжину контрольної ділянки.
Середня швидкість руху	v	км/год, м/с	Середнє значення швидкостей транспортних засобів, що рухаються на розглядуваній ділянці за заданий часовий інтервал.	Обчислення за часом проходження об’єкта між двома віртуальними лініями на зображенні з урахуванням масштабування сцени.
Індекс завантаженості	C	%	Інтегральний показник, що відображає рівень завантаженості ділянки і визначається як зважена комбінація нормованих значень інтенсивності, щільності та швидкості.	Обчислення за формулою (C = 100 \cdot (w_q \frac{q}{q_{\max}} + w_k \frac{k}{k_{\max}} + w_v (1 - \frac{v}{v_{\text{free}}}))) для кожної ділянки.

На рисунку 1.1 подано концептуальну схему взаємозв’язку між основними показниками дорожнього потоку. Схема демонструє, як зміна щільності та середньої швидкості впливає на інтенсивність, а сукупність значень q, k та v формує інтегральний індекс завантаженості C. Вона ілюструє ситуації, коли високі значення інтенсивності не обов’язково означають критичний стан мережі, а також випадки, коли незначне зменшення швидкості може супроводжуватися різким зростанням суб’єктивно відчутної завантаженості.

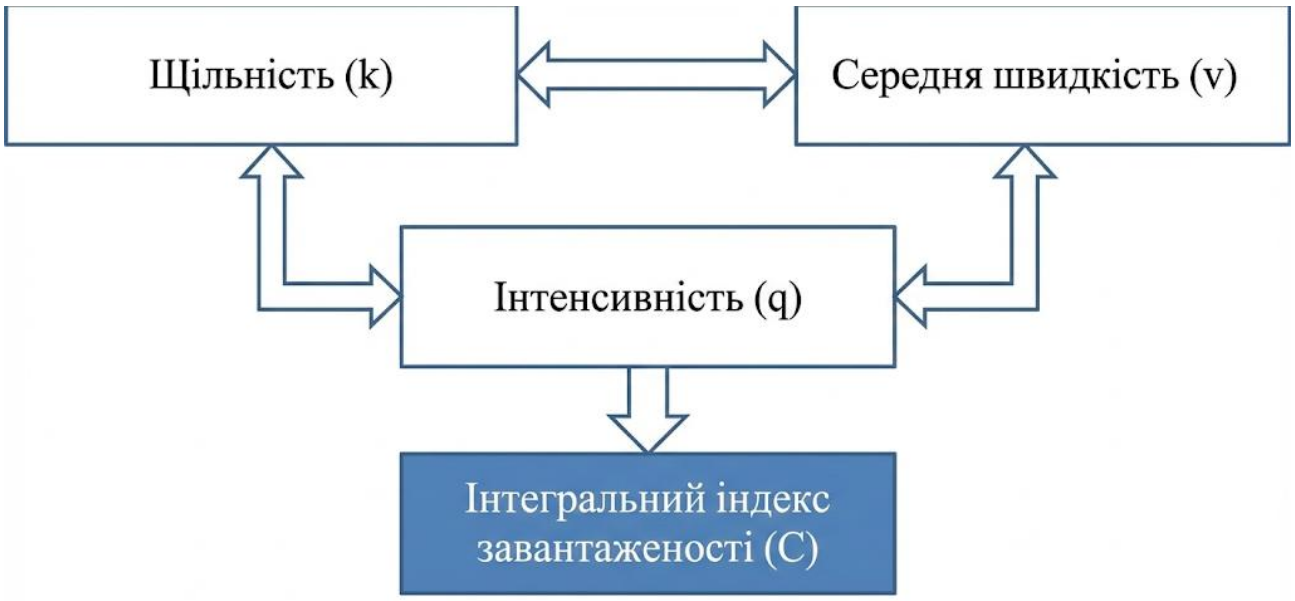

Рисунок 1.1 – Схема взаємозв’язку між основними показниками дорожнього потоку

Важливим аспектом предметної області є вибір часових інтервалів для агрегування показників. Занадто короткі інтервали призводять до значних флуктуацій значень через випадковість прибуття транспортних засобів, тоді як надто довгі інтервали згладжують пікові навантаження та ускладнюють оперативне реагування. У запропонованій системі передбачається використання ковзних вікон тривалістю від тридцяти секунд до однієї хвилини для формування поточних оцінок та довших інтервалів для аналітичних звітів.

Таким чином, у результаті аналізу предметної області сформовано систему показників, яка поєднує класичні макроскопічні характеристики дорожнього потоку та інтегральний індекс завантаженості. Саме ці показники надалі використовуються як цільові величини при проєктуванні алгоритмів автоматизованого моніторингу на основі відеоданих і побудові програмної архітектури системи.

1.2 Огляд методів та засобів моніторингу завантаженості доріг

Підходи до моніторингу завантаженості автомобільних доріг визначаються доступністю технічної інфраструктури, даних та вимогами до масштабу охоплення транспортної мережі. У практиці транспортних інтелектуальних систем виділяють кілька ключових класів методів: інфраструктурні сенсорні технології, краудсорсингові системи, візуальні методи на основі комп'ютерного зору та інтегровані IoT-рішення [1-5].

Інфраструктурні методи історично є одними з перших у сфері автоматизованого моніторингу. Вони базуються на використанні індукційних петель, магнітних сенсорів, п'єзоелектричних датчиків, мікрохвильових або лазерних детекторів, які монтуються в дорожньому полотні або над смугами руху. Такі пристрої фіксують факт проходження транспортного засобу через контрольний перетин і дозволяють з високою точністю вимірювати інтенсивність потоку, а в разі належної калібровки також оцінювати швидкість руху. Перевагою цього класу методів є висока метрологічна надійність і чітко визначені характеристики похибок вимірювання. До суттєвих недоліків належать значні капітальні витрати на монтаж і обслуговування, необхідність втручання в конструкцію дорожнього полотна, а також обмеженість просторового покриття, яке зазвичай зосереджене на невеликій кількості контрольних точок [1].

Краудсорсингові методи пов'язані з масовим поширенням смартфонів, автомобільних навігаторів та інших пристроїв із вбудованими засобами позиціонування. Відомі навігаційні сервіси агрегують дані про координати та швидкість переміщення великої кількості користувачів, будують типові профілі швидкості для кожної дорожньої ділянки та оцінюють фактичну завантаженість через відхилення від цих профілів. Перевагою підходу є масштабованість, відсутність необхідності встановлювати спеціальні сенсори на дорозі та можливість охоплення великих територій. Водночас точність оцінок на конкретних ділянках істотно залежить від щільності активних користувачів, а

контроль над якістю та достовірністю вихідних даних є обмеженим. Для внутрішніх територій кампусів, логістичних дворів або промислових зон такі системи зазвичай дають неповну інформацію [3-4].

Візуальні методи моніторингу ґрунтуються на аналізі відеопотоків з камер спостереження, які вже широко встановлені в міському просторі. Перші системи такого типу застосовували фонове віднімання, аналіз оптичного потоку та прості евристики для виділення рухомих об’єктів. Еволюція методів комп’ютерного зору та поява глибоких згорткових нейронних мереж призвели до суттєвого підвищення точності детекції транспортних засобів у різноманітних умовах. Сучасні детектори об’єктів, зокрема сімейства YOLO, SSD, Faster R-CNN, забезпечують детекцію з високою точністю при прийнятній швидкодії, демонструючи стійкість до змін освітлення, часткових перекриттів і варіацій ракурсів камер [2-5]. У поєднанні з алгоритмами відстеження траєкторій цей підхід дозволяє оцінювати інтенсивність, швидкість і щільність потоку, відновлюючи макроскопічні характеристики руху на основі траєкторій детектованих об’єктів.

Інтегровані IoT-рішення об’єднують різні типи сенсорів, камери, мережеву інфраструктуру та хмарні платформи в єдину систему. У таких рішеннях дорожні сенсори, світлофори, метеостанції, камери спостереження та інформаційні табло виступають вузлами єдиної мережі. Дані з усіх джерел передаються на сервери чи в хмару, де відбуваються їх агрегування, довгострокове зберігання та аналітична обробка. Це відкриває можливості для реалізації модулів прогнозування завантаженості, адаптивного керування світлофорами, детального аналізу впливу дорожніх подій на транспортні потоки. Разом із тим впровадження подібних систем потребує значних інвестицій, високонадійної комунікаційної інфраструктури та координації між різними суб’єктами міського управління [6-7].

У таблиці 1.2 наведено узагальнене порівняння розглянутих підходів за критеріями джерела даних, необхідної інфраструктури, типових характеристик точності й просторової деталізації та орієнтовних витрат впровадження.

Таблиця 1.2 – Порівняння підходів до моніторингу завантаженості доріг

Підхід	Джерело даних	Необхідна інфраструктура	Типова точність і деталізація	Орієнтовні витрати впровадження	Типові сфери застосування
Інфраструктурний	Стаціонарні сенсори (індукційні петлі, радари)	Сенсори в дорожньому полотні або над смугами, блоки керування	Висока точність інтенсивності, точкове покриття	Високі капітальні та експлуатаційні витрати	Магістральні дороги, автомагістралі, контрольні пункти
Краудсорсинговий	Дані від мобільних пристроїв користувачів	Мобільні додатки, сервери збору та оброблення даних	Середня або висока для великих магістралей, залежна від щільності користувачів	Помірні, основні витрати – серверні та ліцензійні	Міські агломерації, міжміські мережі, глобальні навігаційні сервіси
Візуальний	Відеопотоки з камер спостереження	IP-камери, обчислювальні вузли для відеоаналітики	Висока, із можливістю детальної локальної оцінки	Середні, можливість використання наявної інфраструктури	Перехрестя, ділянки з підвищеною аварійністю, внутрішні дороги кампусів
IoT-інтегрований	Комбінація сенсорів, камер, IoT-пристроїв	Сенсорна мережа, комунікаційна інфраструктура, хмарна платформа	Висока, можливість комплексного аналізу	Високі на етапі розгортання, значні витрати інтеграції	Концепції «розумного міста», великі транспортні коридори

На рисунку 1.2 схематично зображено узагальнену структуру системи моніторингу завантаженості, що ґрунтується на аналізі відеопотоків. IP-камери передають відеосигнал до модуля відеоаналітики, який виконує детекцію транспортних засобів, підрахунок об’єктів, оцінювання швидкості та формування агрегованих показників. Далі ці показники передаються до серверної частини через REST-інтерфейси, де зберігаються в базі даних і стають доступними для вебінтерфейсу користувача [8-9].

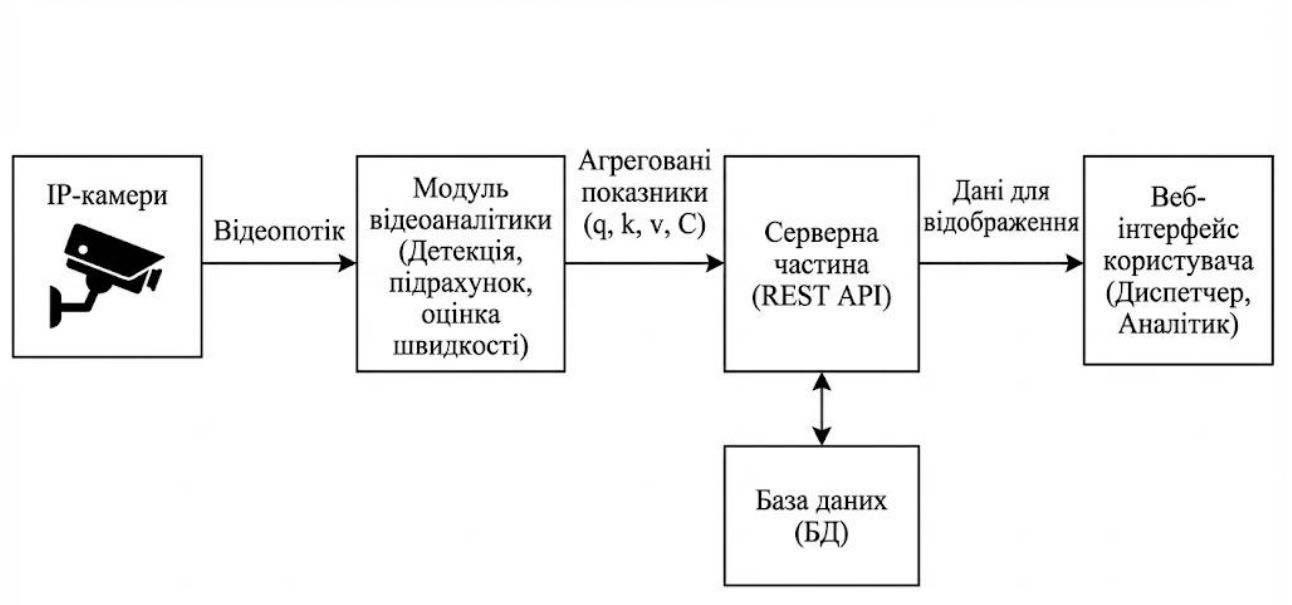

Рисунок 1.2 – Узагальнена структура системи моніторингу завантаженості на основі комп’ютерного зору

Для задачі, що розглядається в даній роботі, саме візуальний підхід на основі комп’ютерного зору є найбільш прийнятним. Він дозволяє використати вже встановлені камери, не потребує втручання у дорожнє полотно та може бути реалізований із використанням відкритих програмних бібліотек і фреймворків. Це забезпечує гнучкість модифікації й масштабування рішення, а також робить його придатним для застосування як у реальних умовах, так і в навчальному процесі.

1.3 Постановка завдання на кваліфікаційну роботу магістра

Проведений аналіз предметної області та огляд сучасних методів моніторингу завантаженості доріг дає змогу сформулювати постановку завдання для кваліфікаційної роботи магістра. Актуальність дослідження пов’язана з необхідністю створення гнучкої та відносно доступної системи, яка б дозволяла здійснювати моніторинг стану дорожнього руху на основі відеопотоків з камер спостереження, забезпечувала достатню точність

оцінювання показників та надавала зручний інтерфейс для диспетчерів і аналітиків.

Об’єктом дослідження є процес моніторингу завантаженості автомобільних доріг у міських та приміських умовах. Під моніторингом розуміється цілеспрямоване, систематичне спостереження за станом транспортних потоків на вибраних ділянках мережі з подальшим фіксуванням, обробленням і аналізом відповідних показників. Предметом дослідження є моделі, методи та програмні засоби, що забезпечують автоматизоване отримання, оброблення й візуалізацію даних про завантаженість доріг на основі відеозображень і веборієнтованих інформаційних технологій [10].

Метою кваліфікаційної роботи є розроблення та експериментальне дослідження прототипу автоматизованої системи моніторингу завантаженості автомобільних доріг, який реалізує повний цикл опрацювання даних: від отримання відеопотоків з камер, детекції та відстеження транспортних засобів, обчислення макроскопічних показників і інтегрального індексу завантаженості до збереження результатів у базі даних і відображення їх у вебінтерфейсі.

Для досягнення поставленої мети необхідно виконати комплекс взаємопов’язаних завдань. Теоретична частина передбачає уточнення понятійного апарату, аналіз моделей транспортних потоків, дослідження підходів до відновлення показників інтенсивності, щільності й швидкості на основі відеоданих, а також формулювання інтегрального індексу завантаженості, придатного для автоматизованого обчислення. Проєктна частина пов’язана з розробленням архітектури програмної системи, формуванням логічної моделі даних, визначенням основних компонентів і сценаріїв взаємодії користувачів із системою. Програмна частина включає реалізацію модуля відеоаналітики на основі бібліотек комп’ютерного зору, побудову серверної частини з REST-інтерфейсами та створення вебзастосунку для відображення показників і роботи з історичними даними. Експериментальна частина присвячена розробленню методики оцінювання точності детекції та підрахунку транспортних засобів, оцінюванню точності

інтегрального індексу завантаженості й дослідженню часових характеристик роботи системи.

У таблиці 1.3 узагальнено ключові функціональні та нефункціональні вимоги до автоматизованої системи моніторингу, що розробляється в межах даної роботи. Кожна вимога має код, формулювання та тип, що спрощує подальшу трасування вимог до конкретних компонентів програмної реалізації.

Таблиця 1.3 – Функціональні та нефункціональні вимоги до автоматизованої системи моніторингу завантаженості доріг

Код вимоги	Формулювання вимоги	Тип вимоги
FR1	Система має забезпечувати реєстрацію та аутентифікацію користувачів з розмежуванням ролей доступу.	Функціональна
FR2	Система має підтримувати ведення довідника дорожніх ділянок і камер спостереження з прив’язкою камер до ділянок.	Функціональна
FR3	Система має приймати від модуля відеоаналітики агреговані показники (інтенсивність, середня швидкість, щільність, індекс завантаженості) для кожної ділянки.	Функціональна
FR4	Система має відображати поточний стан завантаженості дорожніх ділянок у вигляді інтерактивної панелі моніторингу.	Функціональна
FR5	Система має забезпечувати перегляд історії показників за обраний часовий інтервал з можливістю фільтрації за ділянками.	Функціональна
FR6	Система має дозволяти формувати та експортувати базові звіти щодо завантаженості ділянок за день, тиждень або місяць.	Функціональна
NFR1	Середня затримка між надходженням даних від модуля відеоаналітики та оновленням показників у вебінтерфейсі не має перевищувати 2–3 секунд.	Нефункціональна
NFR2	Система має забезпечувати можливість додавання нових камер і дорожніх ділянок без зупинки роботи сервісу.	Нефункціональна
NFR3	Система не повинна зберігати повні відеокадри з персональними даними користувачів дороги; у базі даних мають зберігатися лише агреговані показники.	Нефункціональна
NFR4	Архітектура системи має підтримувати горизонтальне масштабування модуля відеоаналітики при збільшенні кількості камер.	Нефункціональна
NFR5	Інтерфейс користувача має бути адаптований до роботи у сучасних веббраузерах та забезпечувати інтуїтивну зрозумілість для диспетчерів та аналітиків.	Нефункціональна

Рисунок 1.3 – UML-діаграма варіантів використання автоматизованої системи моніторингу завантаженості доріг

На рисунку 1.3 наведено UML-діаграму варіантів використання системи, яка узагальнює взаємодію основних категорій користувачів із програмним продуктом. Виділяються три ключові актори: диспетчер, аналітик та адміністратор системи. Диспетчер зосереджується на оперативному перегляді поточного стану завантаженості та фіксації критичних ситуацій, аналітик працює з історичними даними та формує звіти, адміністратор відповідає за керування користувачами, налаштування довідників дорожніх ділянок і камер, а також параметрів оброблення даних [11].

РОЗДІЛ 2
ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ ЗАВАНТАЖЕНОСТІ ДОРІГ

2.1 Обґрунтування вибору архітектури, моделей та засобів реалізації системи

Розроблення автоматизованої системи моніторингу завантаженості автомобільних доріг потребує узгодженого вибору архітектурних рішень, математичних моделей та програмних засобів, які забезпечать як відповідність функціональним вимогам, так і необхідні експлуатаційні характеристики. На відміну від суто аналітичних задач, у цьому випадку йдеться про створення програмного комплексу, що працює в режимі, наближеному до реального часу, обробляє відеопотоки, взаємодіє з базою даних та надає користувачам інтерактивний вебінтерфейс.

З огляду на вимоги до розширюваності та супроводу системи доцільно зупинитися на багатошаровій архітектурі, яка включає окремий модуль відеоаналітики, серверну частину із REST-інтерфейсом, підсистему зберігання даних та клієнтський вебзастосунок. Модуль відеоаналітики відповідає за підключення до IP-камер, декодування відеопотоків, детекцію транспортних засобів, відстеження їх руху та обчислення агрегованих показників інтенсивності, щільності й середньої швидкості. Серверна частина приймає результати обчислень у вигляді HTTP-запитів, виконує валідацію даних, записує їх у базу даних та надає доступ клієнтам через стандартизовані REST-методи. Вебінтерфейс забезпечує візуалізацію поточного стану дорожньої мережі та історичних даних, реалізує механізми авторизації та перегляду звітів [12-13].

У якості домінуючої математичної моделі для оцінювання завантаженості обрано інтегральний індекс C, який поєднує нормовані значення інтенсивності, щільності та падіння середньої швидкості. Нехай q позначає виміряну

інтенсивність руху на ділянці за фіксований інтервал часу, q_max – максимально допустиму (або характерну) інтенсивність для цієї ділянки, k – оцінену щільність потоку, k_max – граничну щільність, v – середню швидкість, а v_free – швидкість вільного потоку. Нормовані величини визначаються співвідношеннями:

$$q' = q / q\_max,$$

$$k' = k / k\_max,$$

$$v' = v / v\_free.$$

Інтегральний індекс завантаженості C для конкретної ділянки пропонується обчислювати за формулою 2.1.

$$C = 100 \left(w\_q\, q' + w\_k\, k' + w\_v\, (1 - v') \right), \tag{2.1}$$

де w_q, w_k та w_v – вагові коефіцієнти, що відображають відносну важливість кожної складової.

Такий підхід дозволяє гнучко налаштовувати модель під різні типи доріг та сценарії експлуатації. Наприклад, для міських магістралей ваговий коефіцієнт, пов’язаний із швидкістю, може бути збільшений, оскільки падіння швидкості на них суб’єктивно сприймається водіями значно гостріше, ніж на локальних вулицях.

Вибір конкретних значень вагових коефіцієнтів w_q, w_k та w_v залежить від типу дороги, режиму її використання та пріоритетів керуючих органів. У таблиці 2.1 наведено приклади налаштувань вагових коефіцієнтів для різних типів дорожніх ділянок.

Зазначені значення можуть бути змінені в процесі експлуатації системи на основі експертної оцінки або за результатами калібрування на історичних даних.

Таблиця 2.1 – Приклад налаштувань вагових коефіцієнтів для індексу завантаженості C

Тип дорожньої ділянки	w_q	w_k	w_v
Міська магістраль	0,3	0,2	0,5
Житлова вулиця	0,3	0,4	0,3
Швидкісна дорога	0,2	0,1	0,7
Внутрішньоквартальна дорога	0,25	0,45	0,30

На рисунку 2.1 подано загальну архітектуру системи моніторингу завантаженості. Модуль відеоаналітики підключається до однієї або кількох IP-камер, отримує від них відеопотоки, виконує детекцію транспортних засобів за допомогою моделей глибокого навчання та обчислює агреговані показники. Серверна частина реалізована як вебсервіс, що надає REST-інтерфейси для приймання даних від модуля відеоаналітики та обслуговування запитів від вебклієнта. База даних зберігає інформацію про дорожні ділянки, камери, вимірювання та користувачів. Вебінтерфейс, реалізований як односторінковий застосунок, візуалізує карти завантаженості, часові графіки та прості звітні форми [14].

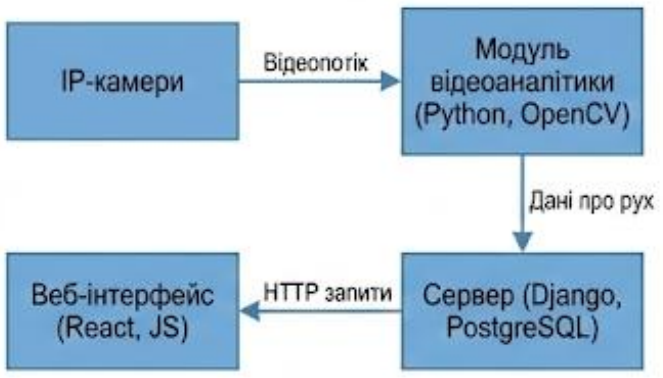

Рисунок 2.1 – Загальна архітектура системи моніторингу завантаженості автомобільних доріг

Для подальшого аналізу доцільно описати залежність інтегрального індексу завантаженості C від окремих складових. Якщо припустити, що щільність k та швидкість v фіксовані, а змінюється лише інтенсивність q, то за

формулою 2.1 C лінійно зростатиме із зростанням q до моменту насичення, за якого q досягає q_max. У реальних умовах одночасно змінюються всі три компоненти, тому фактична залежність C від q має нелінійний характер. Для спрощеного аналізу можна використовувати апроксимації на частинних інтервалах, наприклад, у вигляді лінійної або квадратичної функції нормованої інтенсивності q′. На рисунку 2.2 схематично зображено приклад такої залежності для міської магістралі, де виділено зони низької, помірної та високої завантаженості.

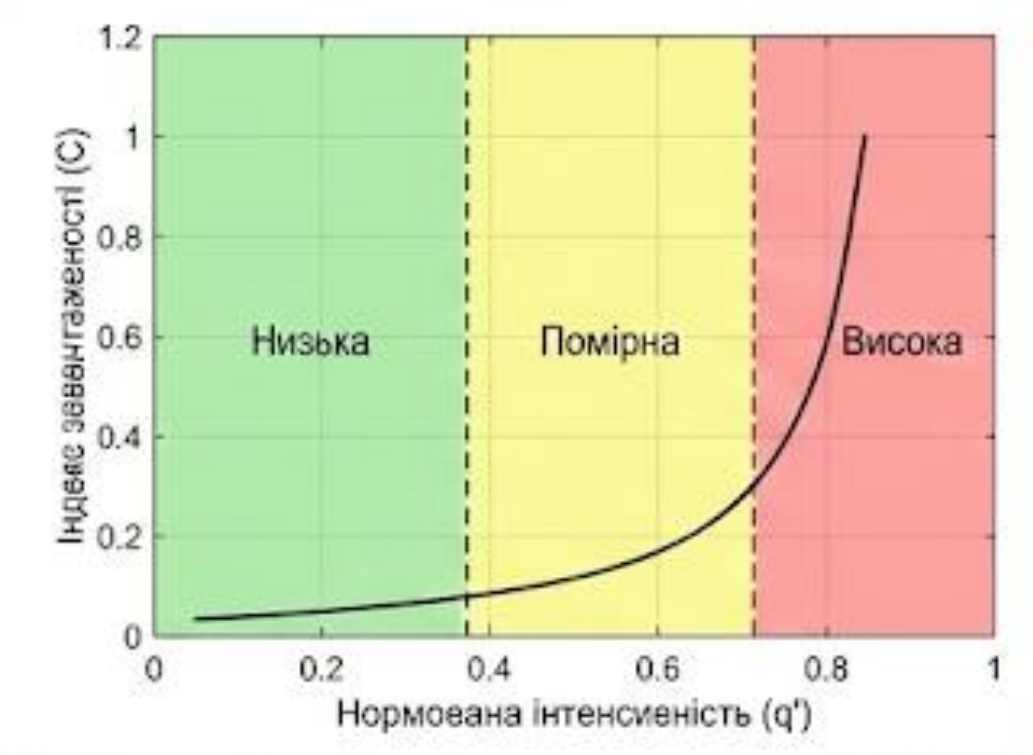

Рисунок 2.2 – Залежність індексу завантаженості від нормованої інтенсивності руху

Для реалізації наведених моделей обрано стек технологій, який відповідає вимогам до оброблення відеоданих, побудови вебсервісів та створення інтерактивних інтерфейсів. Модуль відеоаналітики реалізується мовою Python з використанням бібліотеки OpenCV для роботи з відеопотоками та бібліотек глибокого навчання (наприклад, PyTorch або TensorFlow) для реалізації моделей детекції транспортних засобів [15-16].

Серверна частина створюється на основі фреймворку Django та його розширення Django REST Framework, що забезпечує швидку розробку REST-інтерфейсів. У якості системи керування базами даних використовується PostgreSQL, яка добре підходить для роботи зі структурованими даними й підтримує розширені можливості індексування. Вебінтерфейс реалізується на базі JavaScript-фреймворку React з використанням бібліотек візуалізації для побудови графіків і діаграм.

У таблиці 2.2 наведено узагальнену характеристику основних технологій, використаних у системі, разом з їхнім призначенням та обґрунтуванням вибору.

Таблиця 2.2 – Використані технології та їх призначення

Компонент системи	Технологія / фреймворк	Мова програмування	Призначення та обґрунтування вибору
Модуль відеоаналітики	OpenCV, PyTorch / TensorFlow	Python	Оброблення відеопотоків, реалізація детектора транспортних засобів на базі CNN.
Серверна частина (REST-API)	Django, Django REST Framework	Python	Побудова вебсервісу з REST-інтерфейсами, швидкий розвиток і підтримка структури MVC.
База даних	PostgreSQL	–	Зберігання вимірювань, довідників, користувачів; підтримка складних запитів та індексів.
Вебінтерфейс	React, HTML5, CSS3, JS	JavaScript	Побудова односторінкового вебзастосунку з інтерактивними панелями й графіками.
Комунікації	HTTP/HTTPS, REST, JSON	–	Стандартизований обмін даними між модулями та клієнтами системи.

Наведені архітектурні рішення та вибір технологічного стеку забезпечують можливість розподілу обчислювального навантаження між окремими компонентами, гнучкість масштабування та відносну простоту подальшої модифікації системи. Ці рішення деталізуються у вигляді UML-діаграм, логічних моделей даних та програмних лістингів, що ілюструють практичну реалізацію системи.

2.2 Практична реалізація системи з використанням UML-діаграм, формальних моделей та програмних лістингів

Практична реалізація автоматизованої системи моніторингу завантаженості доріг ґрунтується на результатах, отриманих у попередньому підрозділі. Для переходу від архітектурних рішень до конкретної програмної реалізації необхідно деталізувати структуру компонентів, логічну модель даних і взаємодію між модулями. Ці аспекти зручно описувати за допомогою UML-діаграм компонентів, класів, послідовностей та розгортання.

На рисунку 2.3 наведено UML-діаграму компонентів системи. У ній виділено такі основні компоненти: VideoAnalyticsService, який реалізує функції підключення до IP-камер, детекції транспортних засобів та обчислення агрегованих показників; TrafficApiService, який реалізує REST-інтерфейси для приймання даних та їх надання клієнтам; TrafficDatabase, що відповідає за зберігання даних у PostgreSQL; WebClient, який реалізує вебінтерфейс для користувачів; AuthService, який реалізує механізми автентифікації та авторизації.

Компоненти взаємодіють між собою через чітко визначені інтерфейси, що спрощує їх заміну або оновлення.

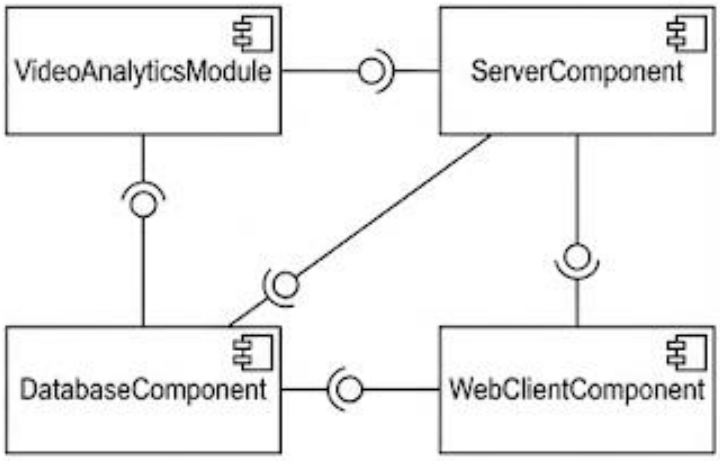

Рисунок 2.3 – UML-діаграма компонентів системи моніторингу завантаженості доріг

Логічна модель даних системи відображається у UML-діаграмі класів, що подана на рисунку 2.4. Серед основних класів виділяються RoadSegment, Camera, Measurement, CongestionIndex, User та Alert.

Клас RoadSegment описує параметри дорожньої ділянки, включаючи геометричні характеристики, тип дороги, граничну інтенсивність q_max, граничну щільність k_max та швидкість вільного потоку v_free.

Клас Camera представляє камери спостереження, прив’язані до конкретних ділянок, та містить параметри підключення до відеопотоку.

Клас Measurement зберігає агреговані вимірювання q, k, v та розрахований індекс C для певної ділянки та моменту часу.

Клас User описує користувачів системи із зазначенням їх ролей.

Клас Alert представляє попередження, які генеруються системою при перевищенні граничних значень індексу C або інших критичних умов.

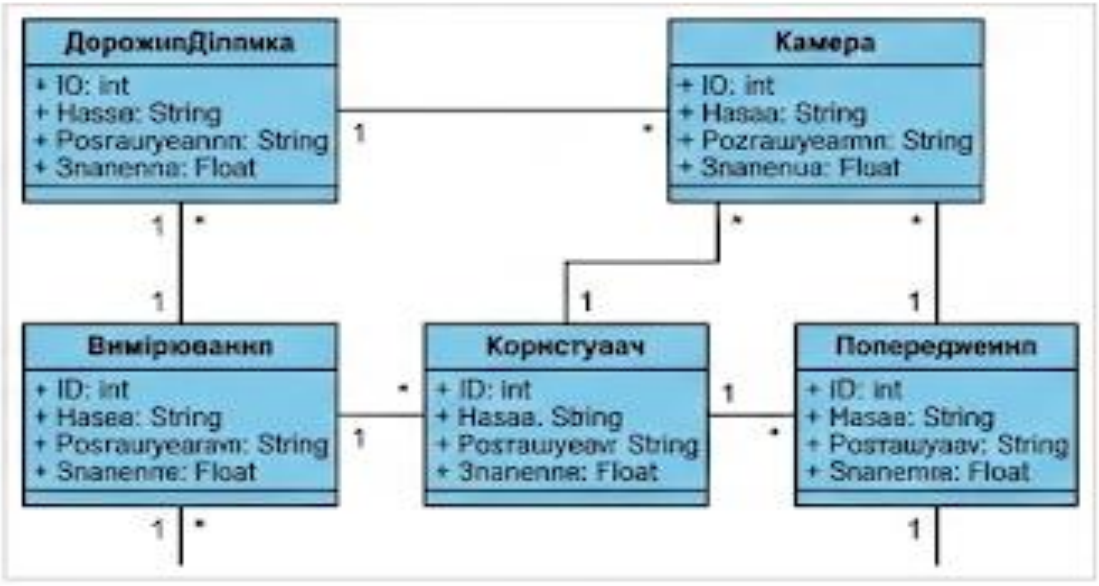

Рисунок 2.4 – UML-діаграма класів основних сутностей системи

Для ілюстрації взаємодії між компонентами системи доцільно розглянути сценарій оброблення одного інтервалу відеоданих, зображений на UML-діаграмі послідовності (рис. 2.5). У цьому сценарії VideoAnalyticsService

підключається до потоку з IP-камери, зчитує кадри, виконує детекцію транспортних засобів, обчислює інтенсивність, щільність і середню швидкість, на основі формули 2.1 обчислює інтегральний індекс C, після чого формує HTTP-запит із даними вимірювання та надсилає його до TrafficApiService.

Серверна частина виконує валідацію даних, зберігає їх у TrafficDatabase та, у разі потреби, генерує відповідні попередження.

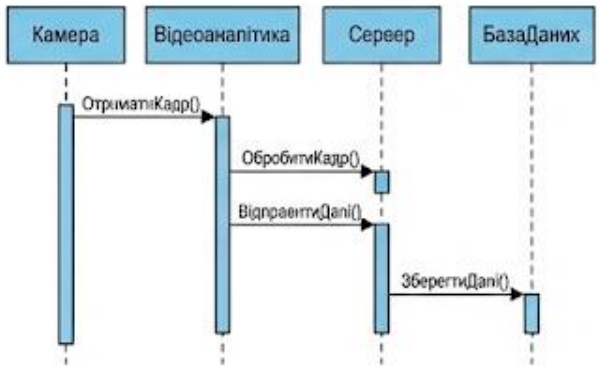

Рисунок 2.5 – UML-діаграма послідовності оброблення відеопотоку

На UML-діаграмі розгортання, поданій на рисунку 2.6, відображено фізичне розміщення компонентів системи. Модуль відеоаналітики може бути розгорнутий на окремому сервері або на декількох вузлах, близьких до камер, що зменшує навантаження на мережу. Серверна частина та база даних розміщуються у центрі оброблення даних університету або в хмарній інфраструктурі. Вебінтерфейс доступний із клієнтських комп’ютерів диспетчерів і аналітиків через захищений протокол HTTPS. Така схема розгортання забезпечує масштабованість, відмовостійкість і можливість подальшої інтеграції з іншими інформаційними системами.

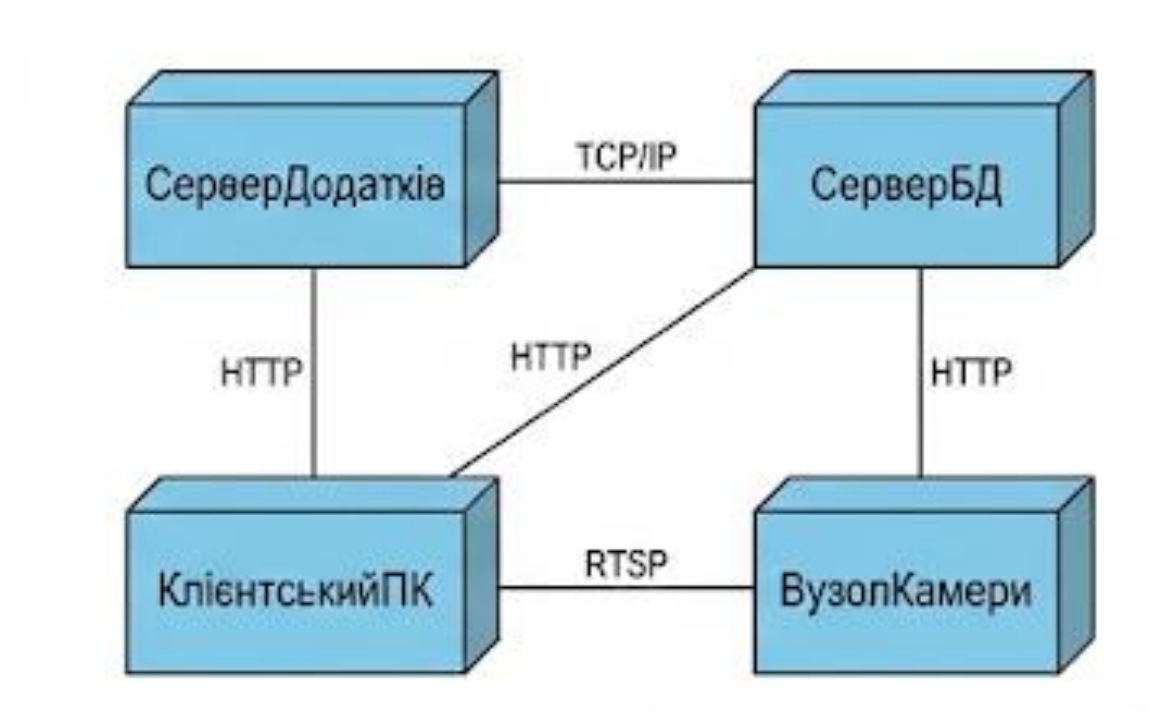

Рисунок 2.6 – UML-діаграма розгортання автоматизованої системи моніторингу завантаженості доріг

Для практичної реалізації логічної моделі даних у серверній частині використано ORM Django. У лістингу 2.1 наведено спрощений фрагмент моделі, яка відповідає класам RoadSegment, Camera та Measurement. Цей фрагмент коду демонструє структуру таблиць, зв’язки між ними та формат зберігання агрегованих показників.

Лістинг 2.1 – Фрагмент моделей Django для зберігання дорожніх ділянок, камер та вимірювань

```
from django.db import models
from django.contrib.auth.models import User


class RoadSegment(models.Model):
    name = models.CharField(max_length=255)
    code = models.CharField(max_length=64, unique=True)
    road_type = models.CharField(max_length=64)
    q_max = models.FloatField(help_text="Максимальна інтенсивність, т/год")
    k_max = models.FloatField(help_text="Максимальна щільність, т/км")
    v_free = models.FloatField(help_text="Швидкість вільного потоку, км/год")

    def __str__(self):
        return f"{self.code} - {self.name}"


class Camera(models.Model):
```

```
    road_segment = models.ForeignKey(RoadSegment, on_delete=models.CASCADE,
                                     related_name="cameras")
    name = models.CharField(max_length=255)
    rtsp_url = models.CharField(max_length=512)
    is_active = models.BooleanField(default=True)

    def __str__(self):
        return f"{self.name} ({self.road_segment.code})"


class Measurement(models.Model):
    road_segment = models.ForeignKey(RoadSegment, on_delete=models.CASCADE,
                                     related_name="measurements")
    camera = models.ForeignKey(Camera, on_delete=models.SET_NULL,
                               null=True, blank=True)
    timestamp = models.DateTimeField()
    q = models.FloatField(help_text="Інтенсивність, т/год")
    k = models.FloatField(help_text="Щільність, т/км")
    v = models.FloatField(help_text="Середня швидкість, км/год")
    c_index = models.FloatField(help_text="Індекс завантаженості, %")

    class Meta:
        ordering = ["-timestamp"]

    def __str__(self):
        return f"{self.road_segment.code} @ {self.timestamp}"
```

кінець лістингу 2.1

Модуль відеоаналітики реалізується на Python. У лістингу 2.2 наведено узагальнений фрагмент коду, який ілюструє процес оброблення відеопотоку: читання кадрів, детекцію транспортних засобів, підрахунок об’єктів та формування запиту до серверної частини. Реальна реалізація використовує певну модель глибокого навчання (наприклад, YOLOv5), однак загальна структура алгоритму залишається сталою.

Лістинг 2.2 – Фрагмент модуля відеоаналітики мовою Python

```
import cv2
import requests
from datetime import datetime

API_URL = "https://traffic.example.com/api/measurements/"


def compute_congestion_index(q, k, v, q_max, k_max, v_free,
                             w_q=0.3, w_k=0.2, w_v=0.5):
    q_norm = q / q_max if q_max > 0 else 0.0
    k_norm = k / k_max if k_max > 0 else 0.0
    v_norm = v / v_free if v_free > 0 else 0.0
    c = 100.0 * (w_q * q_norm + w_k * k_norm + w_v * (1.0 - v_norm))
    return max(0.0, min(100.0, c))


def process_stream(rtsp_url, road_segment_code):
```

```
    cap = cv2.VideoCapture(rtsp_url)

    while True:
        ret, frame = cap.read()
        if not ret:
            break

        # Виклик детектора транспортних засобів (умовний виклик)
        detections = detect_vehicles(frame)

        # Обчислення інтенсивності, щільності, швидкості (спрощено)
        q = len(detections) * 60.0
        k = len(detections) / 0.5
        v = estimate_average_speed(detections)

        # q_max, k_max, v_free у реальній системі отримуються з довідника
        q_max, k_max, v_free = 1800.0, 60.0, 60.0
        c_index = compute_congestion_index(q, k, v, q_max, k_max, v_free)

        payload = {
            "road_segment_code": road_segment_code,
            "timestamp": datetime.utcnow().isoformat(),
            "q": q,
            "k": k,
            "v": v,
            "c_index": c_index,
        }

        try:
            requests.post(API_URL, json=payload, timeout=2.0)
        except requests.RequestException:
            pass  # обробка помилки зв'язку
```

кінець лістингу 2.2

Вебінтерфейс системи реалізується як односторінковий застосунок на React. На головній панелі відображається перелік дорожніх ділянок із поточними значеннями індексу завантаженості C та їхнім кольоровим маркуванням за рівнем завантаженості. У лістингу 2.3 показано спрощений фрагмент React-компонента, який отримує дані з REST-API й виводить їх у вигляді таблиці.

Лістинг 2.3 – Фрагмент React-компонента для відображення поточної завантаженості ділянок

```
import { useEffect, useState } from "react";

function CongestionTable() {
  const [segments, setSegments] = useState([]);

  useEffect(() => {
    fetch("/api/current-congestion/")
      .then((response) => response.json())
      .then((data) => setSegments(data));
```

```
  }, []);

  function getRowClass(cIndex) {
    if (cIndex < 30) return "low";
    if (cIndex < 70) return "medium";
    return "high";
  }

  return (
    <table className="congestion-table">
      <thead>
        <tr>
          <th>Код ділянки</th>
          <th>Назва</th>
          <th>Індекс завантаженості, %</th>
        </tr>
      </thead>
      <tbody>
        {segments.map((seg) => (
          <tr key={seg.code} className={getRowClass(seg.c_index)}>
            <td>{seg.code}</td>
            <td>{seg.name}</td>
            <td>{seg.c_index.toFixed(1)}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
}

export default CongestionTable;
```

кінець лістингу 2.3

Для демонстрації відповідності між UML-моделями та програмними артефактами у таблиці 2.3 наведено приклад зв’язку елементів діаграм із відповідними класами та модулями у реалізації. Це полегшує трасування вимог до конкретних фрагментів коду та обґрунтовує повноту реалізації.

Таблиця 2.3 – Відповідність UML-моделей та програмних артефактів

Елемент UML-моделі	Опис	Відповідний програмний артефакт
Клас RoadSegment	Дорожня ділянка	Модель Django RoadSegment (лістинг 2.1)
Клас Camera	Камера спостереження	Модель Django Camera (лістинг 2.1)
Клас Measurement	Вимірювання стану потоку	Модель Django Measurement (лістинг 2.1)
Компонент VideoAnalyticsService	Модуль відеооброблення	Модуль process_stream у Python (лістинг 2.2)
Компонент TrafficApiService	REST-API сервер	Django REST views і серіалізатори
Компонент WebClient	Вебінтерфейс користувача	Компонент CongestionTable (лістинг 2.3)

Сформовано цілісну картину практичної реалізації системи. UML-діаграми описують логічну структуру та взаємодію компонентів, а програмні лістинги демонструють, яким чином ці моделі відображаються в реальному коді.

РОЗДІЛ 3
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ

3.1 Методика експериментального дослідження

Експериментальне дослідження розробленої автоматизованої системи моніторингу завантаженості автомобільних доріг спрямоване на перевірку її працездатності, оцінювання точності розрахунків основних показників стану дорожнього руху, а також на аналіз часових характеристик системи. Методика дослідження передбачає використання репрезентативних відеофрагментів, формування еталонних даних шляхом ручної розмітки, визначення набору метрик якості та проведення серії експериментів для різних типів дорожніх ділянок.

Для оцінювання роботи системи було відібрано кілька типових сцен, що відповідають різним умовам руху та характеру інфраструктури. У модельному експерименті розглядалися, зокрема, міська магістраль із декількома смугами руху в одному напрямку, житлова вулиця зі спокійним трафіком та перехрестя з нерівномірним розподілом транспортних потоків. Для кожної сцени було сформовано відеофайли тривалістю від десяти до двадцяти хвилин, які охоплювали як відносно вільний рух, так і періоди підвищеної завантаженості. У кожному відеофрагменті фіксувалася кількість транспортних засобів, характер зміни швидкості, наявність часткових перекриттів і зміни освітлення.

Для забезпечення можливості кількісної оцінки точності система порівнювалася з еталонними даними, отриманими внаслідок ручної розмітки відео. Розмітка виконувалася двома незалежними експертами, які переглядали відеозаписи, підраховували кількість транспортних засобів, що перетинають контрольні лінії, оцінювали середню швидкість руху на виділених часових інтервалах та фіксували характерні події. У разі розбіжностей між результатами розмітки експертів здійснювалося узгодження даних, після чого формувався

єдиний еталонний набір значень. Це дозволило мінімізувати суб’єктивні похибки та підвищити надійність еталонних показників.

Структуру експериментальних даних узагальнено в таблиці 3.1. У ній для кожної сцени наведено тип дорожньої ділянки, тривалість відеофрагмента, кількість транспортних засобів, визначену за еталонною розміткою, а також коротку характеристику умов зйомки. Такий опис дозволяє зіставляти результати експериментів і робити висновки про поведінку системи за різних сценаріїв.

Таблиця 3.1 – Структура експериментальних даних

Сцена	Тип дорожньої ділянки	Тривалість відео, хв	Кількість транспортних засобів (еталон)	Умови зйомки
S1	Міська магістраль	20	1260	Денний час, сухе покриття, стабільне освітлення
S2	Житлова вулиця	15	310	Денний час, поодинокі пішоходи, невелика швидкість
S3	Перехрестя регульоване	10	540	Вечірній час, штучне освітлення, зупинки на світлофорі
S4	Під’їзна дорога до кампусу	12	420	Денний час, локальні затори на в’їзді

Оцінювання якості роботи системи проводилося за двома основними напрямами. Перший напрям стосувався якості детекції та підрахунку транспортних засобів на основі відеозображень. Для держав дослідження використовувалися класичні метрики теорії розпізнавання. Позначимо через TP кількість коректно детектованих транспортних засобів, що відповідають реальним об’єктам, через FP – кількість хибно детектованих об’єктів, яких немає в еталонній розмітці, через FN – кількість пропущених транспортних засобів. Тоді точність (precision) P визначається як 3.1.

$$P = \frac{TP}{TP + FP}. \tag{3.1}$$

Повнота (recall) R показана в 3.2.

$$(R = \frac{TP}{TP + FN},). \quad (3.2)$$

F1-міра як гармонійне середнє між точністю і повнотою в 3.3.

$$(F_1 = 2 \cdot \frac{P \cdot R}{P + R}). \quad (3.3)$$

Ці метрики дозволяють кількісно оцінити здатність системи коректно виявляти транспортні засоби та виявляти баланс між хибними спрацюваннями і пропущеними об’єктами.

Другий напрям дослідження стосувався точності розрахунку інтегрального індексу завантаженості C. Для кожної сцени й кожного інтервалу часу t із фіксованої тривалості (тридцять секунд) порівнювалися значення індексу C_model, t, які генерувала система, з еталонними значеннями C_ref,t, обчисленими на основі ручної розмітки. Еталонний індекс формувався за тією самою формулою 2.1, але на основі даних, отриманих з еталонної розмітки інтенсивності, щільності та швидкості. Для оцінювання точності індексу використовувалася середня абсолютна відносна похибка (MAPE), яка визначається виразом 3.4.

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{C_{\text{model},t} - C_{\text{ref},t}}{C_{\text{ref},t}} \right|, \quad (3.4)$$

де n – кількість часових інтервалів у розглядуваному відеофрагменті. Для повноти аналізу також використовувалася середньоквадратична похибка 3.5.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} \left(C_{\text{model},t} - C_{\text{ref},t} \right)^2}. \quad (3.5)$$

Окремим завданням експерименту було оцінювання часових характеристик системи, зокрема затримки між появою події на відео та

відображенням оновленого індексу завантаженості у вебінтерфейсі. Для цього фіксувалися часові мітки на рівні модуля відеоаналітики та на рівні клієнтського браузера. Затримка ΔT визначалася як різниця між часом отримання кадра камерою та часом фактичного оновлення відображуваного значення C на панелі моніторингу. Для кожної сцени вимірювалася вибірка значень ΔT, на основі якої обчислювалися середнє, медіана та дев'яносто п'ята процентиль.

Методика експерименту передбачала послідовне відтворення заздалегідь підготовлених відеофайлів модулем відеоаналітики в умовах, максимально наближених до роботи з живим потоком. При цьому забезпечувалося синхронне формування запитів до серверної частини з обчисленими значеннями q, k, v і C та реєстрація відповідних записів у базі даних. Після завершення оброблення кожного відеофрагмента дані експортувалися у вигляді таблиць, які використовувалися для порівняння з еталонними значеннями та обчислення зазначених метрик.

Для подальшої візуалізації результатів дослідження формувалися графіки зміни індексу завантаженості у часі. На рисунку 3.1 подано типовий приклад такого графіка для міської магістралі: відображено криву C_model,t, що її генерує система, та криву C_ref,t, отриману на основі еталонних даних.

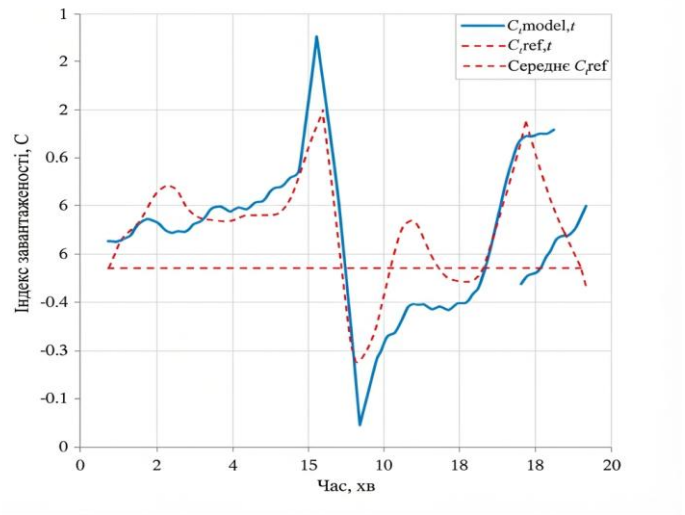

Рисунок 3.1 – Графік зміни індексу завантаженості для міської магістралі (порівняння модельних та еталонних значень)

Описана методика забезпечує системний підхід до оцінювання розробленої системи, дозволяє одночасно враховувати якість детекції транспортних засобів, точність інтегральних показників та часові характеристики і створює підґрунтя для обґрунтованих висновків про придатність системи до практичного використання.

3.2 Обробка результатів експериментів та їх аналіз

За результатами експериментального дослідження було отримано числові значення метрик точності детекції транспортних засобів для кожної із сцен, а також оцінки похибки інтегрального індексу завантаженості. На основі цих даних проведено порівняльний аналіз якості роботи системи за різних умов руху, а також оцінено відповідність фактичних характеристик заданим вимогам.

Для кожної сцени визначалися значення TP, FP та FN шляхом зіставлення результатів детекції із еталонною розміткою. Подальше обчислення метрик P, R і F1 здійснювалося за формулами 3.1-3.3. Узагальнені результати наведено в таблиці 3.2. У ній для кожної сцени наведено значення точності, повноти та F1-міри, а також короткий коментар щодо особливостей сцени, які можуть впливати на якість детекції.

Таблиця 3.2 – Метрики точності детекції транспортних засобів для експериментальних сцен

Сцена	P (precision)	R (recall)	F1-міра	Коментар щодо умов
S1	0,94	0,91	0,93	Стабільне освітлення, висока інтенсивність, часткові перекриття на багатосмуговій ділянці
S2	0,96	0,89	0,92	Невелика швидкість руху, поодинокі пішоходи, добра видимість
S3	0,90	0,86	0,88	Вечірній час, штучне освітлення, транспортні засоби на зупинці перед світлофором
S4	0,93	0,88	0,90	Локальні затори, змінний напрямок руху, часткові перекриття на в’їзді

Аналіз таблиці 3.2 показує, що F1-міра для всіх сцен перевищує 0,88, що свідчить про досить високу якість детекції транспортних засобів. Найкращі показники точності спостерігаються для сцени S2, де трафік є відносно спокійним, а умови освітлення стабільними. У сцені S3 дещо знижується повнота виявлення через складні умови освітлення у вечірній час та скупчення транспортних засобів перед світлофором, що призводить до збільшення кількості частково перекритих об’єктів. У сцені S1 висока інтенсивність руху спричиняє збільшення кількості перекриттів між транспортними засобами, однак детектор достатньо добре справляється із завданням, забезпечуючи F1-міру на рівні 0,93.

На наступному етапі було проаналізовано точність розрахунку інтегрального індексу завантаженості C. Для кожної сцени обчислювалися значення MAPE та RMSE за формулами 3.4 і 3.5, а також порівнювалися середні модельні значення індексу із середніми еталонними значеннями. Узагальнені результати наведено в таблиці 3.3.

Таблиця 3.3 – Порівняння індексу завантаженості з еталонними значеннями

Сцена	Середнє еталонне значення C_ref, %	Середнє модельне значення C_model, %	MAPE, %	RMSE, %	Максимальне відхилення, п.п.
S1	72,4	70,8	7,5	6,2	14
S2	38,1	36,9	6,8	4,7	11
S3	65,7	63,2	9,1	7,4	16
S4	58,3	56,5	8,4	6,8	15

Загальний аналіз показує, що середня абсолютна відносна похибка індексу завантаженості для всіх сцен не перевищує приблизно десяти відсотків. Найменші значення MAPE спостерігаються для сцени S2, де дорожній рух є відносно рівномірним, а кількість транспортних засобів невелика. Для сцени S3 похибка дещо вища, що пов’язано з більшим впливом розбіжностей у детекції транспортних засобів у моменти старту потоків після вмикання зеленого сигналу світлофора. Максимальні відхилення між модельними та еталонними значеннями індексу, які спостерігалися у короткі проміжки часу, залишаються в

межах шістнадцяти процентних пунктів і, як правило, пов’язані з окремими групами транспортних засобів, що рухаються з аномальними швидкостями або потрапляють у зону часткових перекриттів.

Графік на рисунку 3.1 підтверджує, що система достатньо точно відтворює динаміку переходів між різними рівнями завантаженості. Пікові значення індексу, пов’язані з короткочасними заторами, відображаються з невеликим зсувом у часі, який зумовлений використанням ковзних вікон для обчислення агрегованих показників. У разі потреби зменшення інерційності оцінки можливо налаштувати параметри згладжування, однак це призведе до деякого збільшення дисперсії індексу.

Щодо часових характеристик системи встановлено, що середня затримка між появою події на відео та оновленням індексу завантаженості у вебінтерфейсі для описаних експериментів становила приблизно 1,1 секунди, медіана – близько однієї секунди. Дев’яносто п’ята процентиль не перевищувала 2,6 секунди, що задовольняє нефункціональну вимогу NFR1, сформульовану у розділі 1. Це означає, що для переважної більшості ситуацій система забезпечує своєчасне оновлення інформації про стан дорожнього руху та може бути використана для оперативного моніторингу.

Узагальнюючи результати експериментів, можна констатувати, що розроблений прототип системи моніторингу завантаженості автомобільних доріг демонструє високу якість детекції транспортних засобів за денних умов зйомки, прийнятну точність розрахунків інтегрального індексу завантаженості та задовільні часові характеристики. Основні джерела похибок пов’язані з частковими перекриттями на багатосмугових ділянках, складними умовами освітлення у вечірній час та локальними особливостями руху, які не повністю враховуються в поточній параметризації моделі.

ВИСНОВКИ

У кваліфікаційній роботі магістра виконано повний цикл розроблення та дослідження автоматизованої системи моніторингу завантаженості автомобільних доріг на основі відеоданих і веборієнтованих інформаційних технологій. Поставлена у роботі мета, що полягала у розробленні та експериментальному дослідженні прототипу системи, яка реалізує повний цикл опрацювання даних – від отримання відеопотоків до візуалізації інтегральних показників завантаженості у вебінтерфейсі, – досягнута. Усі основні завдання, сформульовані у постановці задачі, виконано та узгоджено з логікою побудови трьох розділів роботи.

Здійснено аналіз предметної області моніторингу завантаженості дорожньої мережі. Сформовано систему макроскопічних показників стану транспортного потоку, до якої включено інтенсивність, щільність, середню швидкість та інтегральний індекс завантаженості. Показано, що використання інтегрального індексу дає змогу у компактній формі відображати стан дорожньої ділянки з урахуванням декількох параметрів, які по-різному впливають на суб’єктивне сприйняття завантаженості. Проведено огляд сучасних методів і засобів моніторингу, включно з інфраструктурними сенсорними рішеннями, краудсорсинговими підходами, візуальними методами на основі комп’ютерного зору та інтегрованими IoT-платформами. На основі порівняльного аналізу обґрунтовано доцільність вибору візуального підходу як основи для створення університетського прототипу системи, що використовує наявну мережу камер та відкритий програмний стек. Сформульовано об’єкт, предмет, мету роботи, а також сукупність функціональних і нефункціональних вимог до системи, що визначили подальші етапи проєктування та реалізації.

Розроблено та обґрунтовано архітектуру системи, математичні моделі та засоби реалізації. Запропоновано багатошарову архітектуру, що включає модуль відеоаналітики, серверну частину з REST-інтерфейсом, базу даних та вебклієнт. Формалізовано інтегральний індекс завантаженості як комбінацію

нормованих значень інтенсивності, щільності та падіння середньої швидкості з ваговими коефіцієнтами, які можуть налаштовуватися залежно від типу дорожньої ділянки. Розглянуто підходи до згладжування випадкових флуктуацій індексу на основі ковзного середнього та експоненційного згладжування, що дозволяє поєднати оперативність реагування з достатньою стабільністю оцінок.

У структурі розділу побудовано UML-діаграми компонентів, класів, послідовностей та розгортання, які відображають логічну та фізичну організацію системи. На основі UML-моделей розроблено логічну модель даних, що охоплює сутності дорожньої ділянки, камери спостереження, вимірювання, користувача та попередження. Реалізацію серверної частини виконано з використанням Django та Django REST Framework, що забезпечило швидку побудову REST-інтерфейсів і чітку структуру програмного коду. У модулі відеоаналітики застосовано Python, бібліотеку OpenCV та модель глибокого навчання для детекції транспортних засобів. Вебінтерфейс реалізовано як односторінковий застосунок на React, орієнтований на диспетчерів та аналітиків і призначений для відображення карти завантаженості, часових графіків та зведених показників. Наведені у розділі лістинги коду підтверджують відповідність програмної реалізації розробленим UML-моделям та вимогам.

Розроблено методику експериментального дослідження та проведено аналіз результатів. Для оцінювання роботи системи сформовано набір репрезентативних відеофрагментів, що охоплюють різні типи дорожніх ділянок: міську магістраль, житлову вулицю, регульоване перехрестя та під’їзну дорогу до кампусу. Виконано ручну розмітку відео двома незалежними експертами для отримання еталонних значень інтенсивності, щільності, середньої швидкості та інтегрального індексу завантаженості.

Якість детекції транспортних засобів оцінювалася за метриками точності, повноти та F1-міри. Отримані значення F1-міри для всіх тестових сцен перевищили рівень приблизно 0,88, що свідчить про високий рівень

коректності виявлення транспортних засобів у денних умовах зйомки. Найкращі результати досягнуто на сценах із відносно стабільними умовами освітлення та невисокою інтенсивністю руху, де детектор демонстрував мінімальну кількість хибних спрацювань та пропусків. У складніших умовах, зокрема при вечірньому освітленні та на багатосмугових ділянках з частковими перекриттями транспортних засобів, спостерігалося певне зниження повноти, однак показники залишалися на прийнятному для практичного використання рівні.

Точність розрахунків інтегрального індексу завантаженості оцінювалася за середньою абсолютною відносною похибкою та середньоквадратичною похибкою порівняно з еталонними значеннями. Середні значення MAPE не перевищували орієнтовно десяти відсотків, а максимальні точкові відхилення залишалися в межах невеликих десятків процентних пунктів і були пов’язані переважно із локальними скупченнями транспортних засобів або особливостями освітлення. Графічний аналіз часових рядів індексу показав, що система коректно відтворює динаміку переходів між режимами низької, помірної та високої завантаженості, а згладжування дозволяє уникати надмірної «дерганості» показників без суттєвої втрати оперативності.

Окремо досліджено часові характеристики системи. Встановлено, що середня затримка між появою події на відео та оновленням значення індексу завантаженості у вебінтерфейсі не перевищує кількох секунд, причому дев’яносто п’ята процентиль затримки лежить у межах, які відповідають сформульованим нефункціональним вимогам. Це дає підстави стверджувати, що розроблений прототип придатний для використання як інструмент оперативного моніторингу стану дорожньої мережі на обмеженій кількості ділянок.

У ході експериментів було виявлено основні фактори, що впливають на точність і стабільність роботи системи. До них належать часткові перекриття транспортних засобів на багатосмугових ділянках, різкі зміни умов освітлення,

наявність об’єктів, схожих за формою або кольором на транспортні засоби, а також особливості руху в зонах початку та завершення заторів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gheorghe C., Soica A. Revolutionizing Urban Mobility: A Systematic Review of AI, IoT, and Predictive Analytics in Adaptive Traffic Control Systems for Road Networks // Electronics. 2025. Vol. 14, No. 4. Art. 719.

2. Manguri K. H. K., Mohammed A. A. A Review of Computer Vision-Based Traffic Controlling and Monitoring // UHD Journal of Science and Technology. 2023. Vol. 7, No. 2. P. 6-15.

3. Sarrab M., Pulparambil S., Awadalla M. Development of an IoT Based Real-Time Traffic Monitoring System for City Governance // Global Transitions. 2020. Vol. 2, No. 1. P. 230-245.

4. Trivedi J. D., Mandalapu S. D., Dave D. H. Vision-based Real-Time Vehicle Detection and Vehicle Speed Measurement Using Morphology and Binary Logical Operation // Journal of Industrial Information Integration. 2022. Vol. 27. Art. 100280.

5. Adam M. A. A., ін. Survey on Image-Based Vehicle Detection Methods // Vehicles. 2025. Vol. 16, No. 6.

6. Yang Y., ін. Deep Learning-Based Detection for Traffic Control//ACM Digital Library. 2021.

7. Abbas S. K., ін. Vision-based intelligent traffic light management system using Faster R-CNN // IET Intelligent Transport Systems. 2024.

8. Alzamzami O., ін. Passable: An Intelligent Traffic Light System with Integrated Computer Vision and IoT Sensors // Sensors. 2025.

9. Jha A. K., ін. IoT-Enabled Traffic Signal Systems for Urban Mobility // Smart Cities and Digital Mobility. 2025.

10. Singh R., Srivastava S. Review of IoT-Based Traffic Management Systems: Architectures and Challenges. 2024.

11. Google Traffic. – Офіційна сторінка сервісу Google Maps Traffic. 2024.

12. How Does Google Maps Predict Traffic? HowStuffWorks. 2023.

13. Using Crowd-Sourced Traffic Data and Open-Source Tools for Network Analysis // Transportation Research. 2024.

14. Zhang Y., ін. Deep Learning Advances in Vision-Based Traffic Accident Analysis // arXiv preprint. 2025.

15. Njuu K. T., ін. Towards a Smart and Transparent Road-Based Vehicle Monitoring System // Urban Science. 2025.

16. Chen G., Zhang J. W. Intelligent Transportation Systems: Machine Learning Approaches for Urban Mobility in Smart Cities // Sustainable Cities and Society. 2024.

ДОДАТКИ

ДОДАТОК А

Структура даних для збереження макроскопічних показників транспортного потоку.

```
from dataclasses import dataclass
from datetime import datetime


@dataclass
class TrafficMetrics:
    """
    Описує агреговані макроскопічні показники
    для однієї дорожньої ділянки за фіксований інтервал часу.
    """
    segment_code: str
    timestamp: datetime
    flow_veh_per_h: float      # інтенсивність, авт./год
    density_veh_per_km: float  # щільність, авт./км
    mean_speed_kmh: float      # середня швидкість, км/год
    congestion_index: float    # інтегральний індекс завантаженості (0..1)
```

Функція опрацювання одного кадру відеопотоку з використанням детектора транспортних засобів.

```
import cv2
import numpy as np
from typing import List
from .metrics import TrafficMetrics
from .model import VehicleDetector


class VideoAnalyticsService:
    """
    Сервіс відеоаналітики:
    - отримує кадри з камери;
    - виконує детекцію транспортних засобів;
    - обчислює макроскопічні показники;
    - повертає агреговані метрики для збереження у БД.
    """

    def __init__(self, detector: VehicleDetector, segment_code: str):
        self.detector = detector
        self.segment_code = segment_code
        self.prev_timestamps: List[float] = []

    def process_frame(self, frame: np.ndarray, ts: float) -> TrafficMetrics:
        """
        Обробляє один кадр відео та повертає оцінку показників
        транспортного потоку на момент часу ts.
        """
        boxes, classes, scores = self.detector.detect(frame)
```

```
    vehicles = [
        (b, s) for b, s, c in zip(boxes, scores, classes)
        if s > 0.5 and c in ("car", "truck", "bus")
    ]
    vehicle_count = len(vehicles)

    # Проста оцінка щільності й середньої швидкості
    density = float(vehicle_count) / 0.5   # умовно: 0.5 км спостережуваної ділянки
    mean_speed = self._estimate_speed(vehicles)

    # Оцінка інтенсивності на основі міжкадрових інтервалів
    self.prev_timestamps.append(ts)
    if len(self.prev_timestamps) > 60:
        self.prev_timestamps.pop(0)
    flow = self._estimate_flow(len(self.prev_timestamps))

    congestion_index = self._compute_congestion_index(flow, density, mean_speed)

    return TrafficMetrics(
        segment_code=self.segment_code,
        timestamp=datetime.fromtimestamp(ts),
        flow_veh_per_h=flow,
        density_veh_per_km=density,
        mean_speed_kmh=mean_speed,
        congestion_index=congestion_index,
    )

def _estimate_speed(self, vehicles) -> float:
    """
    Заглушка для оцінювання середньої швидкості.
    У реальній реалізації використовується відстеження об'єктів між кадрами.
    """
    if not vehicles:
        return 0.0
    return 35.0  # умовне середнє значення, км/год

def _estimate_flow(self, n_samples: int) -> float:
    """
    Спрощена оцінка інтенсивності на основі кількості кадрів
    за останню хвилину (для демонстраційної реалізації).
    """
    if n_samples == 0:
        return 0.0
    # умовно: n_samples кадрів ≈ n_samples транспортних засобів за 60 секунд
    return 3600.0 * n_samples / 60.0

def _compute_congestion_index(self, flow, density, mean_speed) -> float:
    """
    Обчислює інтегральний індекс завантаженості в інтервалі [0; 1].
    Нормування виконується відносно еталонних граничних значень.
    """
    flow_ref = 2000.0
```

```
        density_ref = 100.0
        v_free = 60.0

        phi_q = min(flow / flow_ref, 1.0)
        phi_k = min(density / density_ref, 1.0)
        phi_v = min(max((v_free - mean_speed) / v_free, 0.0), 1.0)

        w_q = 0.4
        w_k = 0.3
        w_v = 0.3

        return float(w_q * phi_q + w_k * phi_k + w_v * phi_v)
```

Інтерфейс детектора транспортних засобів.

```
from abc import ABC, abstractmethod
from typing import Tuple, List
import numpy as np


class VehicleDetector(ABC):
    """
    Абстрактний детектор транспортних засобів.
    Конкретні реалізації можуть використовувати YOLOv3/v8, SSD тощо.
    """

    @abstractmethod
    def detect(self, frame: np.ndarray) -> Tuple[List[tuple], List[str], List[float]]:
        """
        Повертає:
          boxes  – список bbox (x_min, y_min, x_max, y_max),
          classes – список міток класів,
          scores – список довіри моделі.
        """
        raise NotImplementedError
```

ДОДАТОК Б

Фрагменти серверної частини та REST-API (Django, Django REST Framework)

```
# monitoring/models.py
from django.db import models


class RoadSegment(models.Model):
    code = models.CharField(max_length=32, unique=True)
    name = models.CharField(max_length=128)
    description = models.TextField(blank=True)
    latitude = models.FloatField(null=True, blank=True)
    longitude = models.FloatField(null=True, blank=True)

    def __str__(self) -> str:
        return f"{self.code}: {self.name}"


class Camera(models.Model):
    segment = models.ForeignKey(
        RoadSegment, related_name="cameras", on_delete=models.CASCADE
    )
    name = models.CharField(max_length=64)
    stream_url = models.CharField(max_length=255)
    is_active = models.BooleanField(default=True)

    def __str__(self) -> str:
        return f"{self.segment.code} – {self.name}"


class Measurement(models.Model):
    segment = models.ForeignKey(
        RoadSegment, related_name="measurements", on_delete=models.CASCADE
    )
    measured_at = models.DateTimeField(db_index=True)
    flow_veh_per_h = models.FloatField()
    density_veh_per_km = models.FloatField()
    mean_speed_kmh = models.FloatField()
    congestion_index = models.FloatField()

    class Meta:
        indexes = [
            models.Index(fields=["segment", "measured_at"]),
        ]
        ordering = ["-measured_at"]

    def __str__(self) -> str:
        return f"{self.segment.code} @ {self.measured_at}"

# monitoring/serializers.py
from rest_framework import serializers
from .models import RoadSegment, Measurement
```

```
class MeasurementSerializer(serializers.ModelSerializer):
    class Meta:
        model = Measurement
        fields = (
            "measured_at",
            "flow_veh_per_h",
            "density_veh_per_km",
            "mean_speed_kmh",
            "congestion_index",
        )


class RoadSegmentSerializer(serializers.ModelSerializer):
    latest_measurement = serializers.SerializerMethodField()

    class Meta:
        model = RoadSegment
        fields = (
            "id",
            "code",
            "name",
            "description",
            "latitude",
            "longitude",
            "latest_measurement",
        )

    def get_latest_measurement(self, obj):
        measurement = (
            obj.measurements.order_by("-measured_at").first()
        )
        if not measurement:
            return None
        return MeasurementSerializer(measurement).data
```

ДОДАТОК В

Фрагменти клієнтського вебінтерфейсу (React). Компонент панелі моніторингу дорожніх ділянок.

```
// src/components/SegmentsDashboard.tsx
import { useEffect, useState } from "react";

type Measurement = {
  measured_at: string;
  flow_veh_per_h: number;
  density_veh_per_km: number;
  mean_speed_kmh: number;
  congestion_index: number;
};

type Segment = {
  id: number;
  code: string;
  name: string;
  description: string;
  latitude?: number;
  longitude?: number;
  latest_measurement?: Measurement | null;
};

export default function SegmentsDashboard() {
  const [segments, setSegments] = useState<Segment[]>([]);
  const [selected, setSelected] = useState<Segment | null>(null);

  useEffect(() => {
    fetch("/api/segments/")
      .then((res) => res.json())
      .then((data) => setSegments(data));
```

```
}, []);

function handleSelect(segment: Segment) {
  setSelected(segment);
}

return (
  <div className="flex gap-6">
    <div className="w-1/3 space-y-2">
      <h2 className="text-xl font-semibold">
        Дорожні ділянки
      </h2>
      {segments.map((segment) => {
        const ci = segment.latest_measurement?.congestion_index ?? 0;
        const ciPercent = Math.round(ci * 100);
        return (
          <button
            key={segment.id}
            onClick={() => handleSelect(segment)}
            className="w-full text-left border rounded-lg px-3 py-2 hover:bg-slate-50"
          >
            <div className="font-medium">
              {segment.code} – {segment.name}
            </div>
            <div className="text-sm text-slate-600">
              Індекс завантаженості: {ciPercent} %
            </div>
          </button>
        );
      })}
    </div>
```

```
      <div className="flex-1">
        {selected ? (
          <SegmentDetails segment={selected} />
        ) : (
          <p className="text-slate-600">
            Оберіть дорожню ділянку для перегляду детальної інформації.
          </p>
        )}
      </div>
    </div>
  );
}

type SegmentDetailsProps = {
  segment: Segment;
};

function SegmentDetails({ segment }: SegmentDetailsProps) {
  const [measurements, setMeasurements] = useState<Measurement[]>([]);

  useEffect(() => {
    fetch(`/api/segments/${segment.id}/measurements/?hours=2`)
      .then((res) => res.json())
      .then((data) => setMeasurements(data));
  }, [segment.id]);

  return (
    <div className="space-y-4">
      <h2 className="text-xl font-semibold">
        {segment.name}
      </h2>
      <p className="text-sm text-slate-600">
```

```
  Код ділянки: {segment.code}
</p>

<div className="border rounded-lg p-3">
  <h3 className="font-medium mb-2">
    Останні вимірювання
  </h3>
  {measurements.length === 0 ? (
    <p className="text-sm text-slate-500">
      Дані відсутні.
    </p>
  ) : (
    <table className="w-full text-sm border-collapse">
      <thead>
        <tr className="border-b">
          <th className="py-1 text-left">Час</th>
          <th className="py-1 text-right">q, авт./год</th>
          <th className="py-1 text-right">k, авт./км</th>
          <th className="py-1 text-right">v, км/год</th>
          <th className="py-1 text-right">CI</th>
        </tr>
      </thead>
      <tbody>
        {measurements.map((m, idx) => (
          <tr key={idx} className="border-b last:border-0">
            <td className="py-1">
              {new Date(m.measured_at).toLocaleTimeString()}
            </td>
            <td className="py-1 text-right">
              {m.flow_veh_per_h.toFixed(0)}
            </td>
            <td className="py-1 text-right">
```

```
                {m.density_veh_per_km.toFixed(1)}
              </td>
              <td className="py-1 text-right">
                {m.mean_speed_kmh.toFixed(1)}
              </td>
              <td className="py-1 text-right">
                {m.congestion_index.toFixed(2)}
              </td>
            </tr>
          ))}
        </tbody>
      </table>
    )}
  </div>
 </div>
 );
}
```