Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра комп'ютерних систем управління

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Автоматизована система моніторингу відеопотоку дорожнього перехрестя»

Виконав: студент 2-го курсу, групи 2АКІТР-23м спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології та роботехніка
(шифр і назва напряму підготовки, спеціальності)

Ярослав РОЇК
(ім'я та прізвище)

Керівник: д.т.н., проф. каф.КСУ
Марія ЮХИМЧУК
(ім'я та прізвище)
«4» 12 2024 р.

Опонент: доц. каф. АІІТ
Володимир ГАРМАШ
(ім'я та прізвище)
«4» 12 2024 р.

Допущено до захисту
Зав. кафедри КСУ, д.т.н., проф.
В'ячеслав КОВТУН
(ім'я та прізвище)
«4» 12 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти другий (магістерський)
Галузь знань – 17 – Електроніка, автоматизація та електронні комунікації
Спеціальність – 174– Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Освітньо-професійна програма – Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ
Зав. кафедри КСУ

В'ячеслав КОВТУН

"14" жовтня 2024 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
студенту Роїку Ярославу Дмитровичу.
(прізвище, ім'я, по батькові)

1. Тема роботи. Автоматизована система моніторингу відеопотоку дорожнього перехрестя
керівник роботи д.т.н Юхимчук Марія Сергіївна
затверджені наказом ВНТУ від "17" вересня 2024 року №310
2. Термін подання студентом роботи "1" грудня 2024 року
3. Вихідні дані до роботи :

1. Савастру С.В., Стеценко І.В. (2023). Методи обробки даних відеокамер спостереження транспортного руху в реальному часі. Міжвідомчий науково- технічний журнал Адаптивні системи автоматичного управління
2. Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Електронний ресурс] // Режим доступу: https://github.com/ultralytics/ultralytics
3. OpenCV. URL:https://docs.opencv.org/4.x/ (date of access: 02.10.2024)

4. Зміст текстової частини: текстова чатина включає анотацію, зміст, вступ, основну частину з першим, оглядовим, розділом, висновки, список посилань та додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): об'єкт, предмет та мета дослідження, структурна схема запропонованої програмно-апаратної системи, алгоритми роботи програної складової розробки, результати тестванння створеної системи.

1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
4	Кавецький В.В. – доцент кафедри економіки підприємства і виробничого менеджменту		

2. Дата видачі завдання "14" жовтня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналітичний огляд переваг та недоліків, які характеризують досліджуваний технологічний процес	17.09.2024	26.09.2024	
2	Дослідження стану розв'язання технічної задачі	27.09.2024	18.10.2024	
3	Розробка та тестування системи	19.10.2024	01.11.2024	
4	Розрахунок економічної частини	02.10.2024	11.11.2024	
5	Оформлення матеріалів до захисту	12.11.2024	28.11.2024	

Студент Ярослав РОЇК
(підпис) (*Ім'я ПРІЗВИЩЕ*)

Керівник роботи Марія ЮХИМЧУК
(підпис) (*Ім'я ПРІЗВИЩЕ*)

АНОТАЦІЯ

УДК. 681.518

Роїк Я. Д. Автоматизована система моніторингу відеопотоку дорожнього перехрестя. Магістерська кваліфікаційна робота зі спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка**.,** освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2024. - 129 с.

На укр. мові. Бібліогр.: 32 назв; рис.: 8; табл. 23.

Метою магістерської кваліфікаційної роботи є підвищення ефективності аналізу стану дорожнього трафіку в реальному часі за даними з відеокамер спостереження за рахунок розробки алгоритмічного та програмного забезпечення

У оглядово-аналітичній частині роботи розглянуто сучасні методи аналізу дорожнього трафіку, особливості збору та обробки даних відеоспостереження, а також переваги використання таких підходів для оптимізації транспортної інфраструктури. Проведено аналіз існуючих рішень для оцінки інтенсивності та завантаженості руху. Описано їх обмеження та перспективи вдосконалення.

У процесі дослідження вдосконалено метод розрахунку показника завантаженості TLCR (Traffic Line Congestion Ratio) та запропоновано новий показник TLIR (Traffic Line Intensity Ratio), який оцінює інтенсивність руху. Розроблено програмний засіб на базі обчислювальних алгоритмів для аналізу відеоданих у реальному часі.

Ключові слова: автоматизація, моніторинг, дорожній рух, відеопотік, система, перехрестя, відеоспостереження.

ABSTRACT

UDC. 681.518

Roik Ya. D. Automated system for monitoring video flow of a road intersection. Master's qualification work in specialty 174 – Automation, computer-integrated technologies and robotics., educational program – Intelligent computer systems. Vinnytsia: VNTU, 2024. - 129 p.

In Ukrainian. Bibliography: 32 titles; fig.: 8; table. 23.

The purpose of the master's qualification work is to increase the efficiency of analyzing road traffic conditions in real time using data from surveillancecameras through the development of algorithmic and software.

In the review and analytical part of the work, modern methods of road traffic analysis, features of collecting and processing video surveillance data, as well as the advantages of using such approaches for optimizing transport infrastructure, are considered. An analysis of existing solutions for assessing traffic intensity and congestion has been conducted. Their limitations and prospects for improvement are described.

In the process of research, the method for calculating the TLCR (Traffic Line Congestion Ratio) congestion index has been improved and a new TLIR (Traffic Line Intensity Ratio) indicator has been proposed, which assesses traffic intensity. A software tool based on computational algorithms for analyzing video data in real time has been developed.

Keywords: automation, monitoring, traffic, video stream, system, intersection, video surveillance.

Зміст

ВСТУП 4
1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ 7
1.1 Опис предметної області 7
1.2 Методи аналізу дорожнього трафіку 9
1.3 Методи виявлення транспортних засобів 14
1.4 Методи визначення динамічних характеристик дорожнього трафіку 17
1.5 Методи визначення статичних характеристик дорожнього трафіку 21
1.6 Постановка задачі та висновки до розділу 23
2 РОЗРОБКА МЕТОДІВ ОЦІНКИ ЩІЛЬНОСТІ ТА ІНТЕНСИВНОСТІ ТРАФІКУ 25
2.1 Модифікований показник TLCR 25
2.2 Показник інтенсивності дорожнього трафіку Traffic Lane Intensity Ratio (TLIR) 32
2.3 Система показників для визначення поточного стану трафіку на ділянці дороги 37
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 40
3.1 Моделювання програмного засобу для обчислення показників MTLCR та TLIR 40
3.2 Інструменти для розробки програмного засобу 43
3.3 Реалізація програмного засобу 44
3.3.1 Модуль обчислення показника MTLCR 46
3.3.2 Реалізація модуля обчислення середньої швидкості потоку 51
3.3.3 Реалізація модуля обчислення показника TLIR 55
3.3.4 Розробка модуля конфігурації та моніторингу 56
4 ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНОГО РІШЕННЯ 69
4.1 Оцінка ефективності обчислення MTLCR 69
4.2 Оцінка ефективності обчислення TLIR 71
5 ЕКОНОМІЧНА ЧАСТИНА 73

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки 73
5.2 Розрахунок узагальненого коефіцієнта якості розробки 76
5.3 Розрахунок витрат на проведення науково-дослідної роботи 78
5.3.1 Витрати на оплату праці 78
5.3.2 Відрахування на соціальні заходи 81
5.3.3 Сировина та матеріали 81
5.3.4 Розрахунок витрат на комплектуючі 83
5.3.5 Спецустаткування для наукових (експериментальних) робіт 83
5 3.6 Програмне забезпечення для наукових (експериментальних) робіт 84
5.3.7 Амортизація обладнання, програмних засобів та приміщень 85
5.3.8 Паливо та енергія для науково-виробничих цілей 86
5.3.9 Службові відрядження 87
5.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації 88
5.3.11 Інші витрати 88
5.3.12 Накладні (загальновиробничі) витрати 88
ВИСНОВКИ 95
ПЕРЕЛІК ПОСИЛАНЬ 97
Додатки 101
ДОДАТОК А (обов’язковий). ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ 102
ДОДАТОК Б (обов’язковий). Технічне завдання 103
ДОДАТОК В (обов’язковий). Лістинг програмного коду 106
ДОДАТОК Г (обов’язковий). Ілюстративна частина 118

ВСТУП

Актуальність теми. В сучасному світі, неможливо уявити велике місто без розвиненої транспортної інфраструктури, невід'ємною частиною якої є автомобільні дороги, що є "артеріями", які дозволяють підтримувати життя в місті. Відповідно оптимальність планування даних "артерій" є критично важливою для розвитку будь- яких населених пунктів. Помилки в даному процесі можуть призвести до недостатньої пропускної здатності ділянки дороги, або ж до надмірних витрат на інфраструктуру ділянок доріг, що використовуються менш активно. І це стосується не тільки пропускної здатності та неоптимальних фінансових витрат, але й безпеки та комфорту усіх учасниіків дорожнього руху. Тож дуже важливо приймати рішення щодо розвитку дорожньої системи, спираючись на факти, а не лише на припущення. А для отримання об'єктивних фактів та прогнозів, необхідно мати методи, що їх надаватимуть. Додатково ці методи можна було б використовувати для задач менших масштабів, таких як регулюванняінтервалів світлофорів або ж часу роботи реверсивних полос руху.

Існує багато різноманітних методів збору даних та оцінки стану трафіку за допомогою спеціалізованих сенсорів та відеокамер, проте не має стандартизованих методів для всебічного аналізу трафіку. Існуючі методи зосереджуються лише на одному з параметрів: інтенсивності або ж завантаженості дороги. Наприклад, на підрахунку кількості автомобілів, що проїхала ділянку дороги за заданий проміжок часу, або на оцінці кількості автомобілів на ділянці дороги у момент часу. Причому значення оцінок при такому розрахунку матимуть абсолютне значення. Тобто, вони матимуть сенс лише у прив'язці до параметрів досліджуваної ділянки дороги, що не завжди є зручним. Додатково існує ще безліч характеристик трафіку, окрім завантаженості та інтенсивності, які можуть бути корисні для подальшого прийняття рішень щодо розвитку транспортної інфраструктури.

Мета дослідження – підвищити ефективність (точність, швидкодію) існуючих програмних методів аналізу стану дорожнього трафіку за даними з відеокамер спостереження у реальному часі.

Для досягнення даної мети необхідно вирішити наступні завдання:

- вдосконалити показник завантаженості TLCR;

- розробити метод визначення інтенсивності дорожнього руху за даними з відеокамер;

- розробити програмний засіб, що обчислюватиме систему показників завантаженості та інтенсивності за даними з відеокамер у реальному часі.

Об’єкт дослідження – методи та засоби аналізу стану дорожнього трафіку за даними з відеокамер у реальному часі.

Предмет дослідження – методи та програмні засоби обробки даних відеокамер спостереження транспортного руху в реальному часі.

Наукова новизна отриманих результатів магістерської дисертації:

- вдосконалено метод розрахунку показника TLCR (модифікований показник MTLCR), за рахунок усунення впливу ширини транспортних засобів та перспективи на значення показника, внаслідок чого вдалося збільшити точність розрахунку показника на ділянках дороги більших розмірів.

- вперше введено показник TLIR (Traffic Line Intensity Ration) для оцінки інтенсивності руху на ділянці дороги на основі обробки даних відеоряду з використанням розрахованих значень показника MTLCR та актуальної середньої швидкості транспортних засобів на ділянці дороги.

- вперше сформовано систему показників MTLCR та TLIR, що дозволяє повноцінно описувати стан трафіку на ділянці дороги.

Практичне значення отриманих результатів. Отримані результати дозволяють оцінювати обидві складові трафіку: статичну у вигляді показника MTLCR та динамічну у вигляді показника TLIR. Розроблено програмнийзасіб, який дозволяє конфігурувати процеси оцінки стану трафіку, використовуючи дані з камер відеоспостереження, запускати дані процеси на

виконання та отримувати результати у вигляді обчислених показників MTLCR та TLIR, що може бути корисним для подальшого аналізу, моделювання та прогнозування стану трафіку.

Особистий внесок здобувача. Робота містить результати самостійних досліджень здобувача, в яких вкладено авторський підхід до розробки програмних методів для аналізу дорожнього трафіку за даними з камер відеоспостереження. Наукові положення та основні результати, що містяться в дисертації були отримані здобувачем самостійно у процесі науково-дослідної роботи.

Апробація результатів роботи. Основні положення й результати дослідження були обговорення на науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (2024). **Публікації.** Ярослав Дмитрович Роїк «АВТОМАТИЗАЦІЯ СИСТЕМИ КОТРОЛЮ ТРАФІКУ БІЛЯ ПІШОХОДНИХ ПЕРЕХОДІВ»
«Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)», 2024. URL: https://conferences.vntu.edu.ua

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ

1.1 Опис предметної області

Управління дорожньо-транспортною інфраструктурою завжди було складною задачею, що включала не лише підтримку існуючих складових системи, а й планування розвитку та вдосконалення цієї системи. Важливим аспектом такої роботи є використання об'єктивних фактів та прогнозів для прийняття рішень щодо вдосконалення транспортних артерій. Аналіз трафіку та інфраструктури є ключовим елементом для забезпечення ефективної роботи транспортної системи, що забезпечує безпеку, комфорт та економічну ефективність. Правильне управління дорожнім рухом безпосередньо впливає на зменшення заторів, скорочення часу в дорозі, а також на зниження негативного впливу на довкілля.

Історично для аналізу трафіку використовувалися спеціалізовані сенсори та пристрої, які вимірювали лише один параметр (наприклад, кількість автомобілів або швидкість руху). З розвитком технологій з'явилася можливість збору та аналізу більш комплексної інформації, що дозволяє оцінювати кілька параметрів одночасно. Зокрема, технології комп'ютерного зору та нейронні мережі дозволили з новою силою впроваджувати аналіз даних з відеокамер для моніторингу дорожнього руху[1,9].

На відміну від традиційних методів, що використовували датчики або магнітні петлі для фіксації транспортних засобів, камери відеоспостереження надають можливість отримувати більш точні та багатогранні дані. Вони дозволяють не лише фіксувати рух автомобілів, але й виявляти порушення правил дорожнього руху, наприклад, перевищення швидкості, порушення дорожніх знаків або аварійні ситуації. Крім того, відеокамери дозволяють здійснювати контроль за пішоходами, велосипедистами та іншими учасниками дорожнього руху, що дає змогу створити більш комплексну картину для аналізу ситуації на дорозі.

Застосування комп'ютерного зору стало важливим етапом у розробці систем для аналізу трафіку, оскільки зображення та відео з камер спостереження містять величезну кількість корисної інформації. Сучасні методи обробки відеоданих дозволяють автоматично виявляти та відслідковувати транспортні засоби, визначати інтенсивність руху, оцінювати швидкість та інші параметри трафіку, що допомагає у прийнятті оперативних рішень щодо регулювання руху. Завдяки використанню алгоритмів машинного навчання та глибокого навчання, ці системи стали здатні точно визначати типи транспортних засобів, розпізнавати номерні знаки, а також прогнозувати можливі ускладнення руху, наприклад, на підставі історичних даних або поточної ситуації [2].

Розвиток обчислювальних потужностей і алгоритмів машинного навчання дозволив суттєво прискорити і вдосконалити процеси обробки відеоданих. У минулому, для того, щоб здійснити необхідний аналіз трафіку за допомогою відеоспостереження, необхідно було витрачати багато часу на перегляд і аналіз великої кількості відеозаписів вручну. Сучасні методи комп'ютерного зору дозволяють автоматизувати цей процес, зменшуючи потребу в людських ресурсах та підвищуючи ефективність. Це також дає змогу здійснювати моніторинг в реальному часі, що є важливим для оперативного реагування на надзвичайні ситуації або аварії.

Особливе значення ці методи мають у сферах, де використовується камери для спостереження за трафіком на дорогах. Раніше використовувані спеціалізовані сенсори вимірювали лише одну характеристику, але відеокамери дають змогу отримати більш повну картину ситуації на дорозі, включаючи інтенсивність руху, швидкість, типи транспортних засобів тощо. Це дозволяє створювати більш точні прогнози щодо майбутнього стану трафіку, що в свою чергу дає змогу вживати превентивних заходів для покращення ситуації на дорогах.

Завдяки таким інноваційним технологіям виникла нова парадигма в управлінні транспортними системами — Інформаційно-транспортні системи

(ІТС), які активно використовуються для збору, обробки та аналізу даних про трафік. ІТС дозволяють автоматично приймати рішення, наприклад, для регулювання світлофорів в реальному часі залежно від стану трафіку. Вони здатні оперативно реагувати на зміни ситуації на дорогах, що значно зменшує час реагування та підвищує рівень безпеки. Окрім того, ці системи можуть передбачати можливі затори або інші ускладнення руху, що дозволяє зменшити ймовірність аварій та знижує загальну напругу в транспортному потоці.

Світові тенденції зараз спрямовані на створення динамічних адаптивних ІТС, які здатні автоматично налаштовуватися в залежності від ситуації на дорозі, оптимізуючи транспортні потоки та забезпечуючикомфорт і безпеку для всіх учасників руху[6]. Ці системи дозволяють більш ефективно використовувати існуючу інфраструктуру, знижуючи затори і забезпечуючи сталий розвиток транспортної системи. Оскільки більшість міст стикаються з проблемою зростання кількості транспортних засобів, автоматизація і адаптація таких систем стане ключем до збереження високої якості життя для мешканців міських агломерацій.

1.2 Методи аналізу дорожнього трафіку

Серед існуючих параметрів, що описують стан дорожнього трафіку, можна виділити два основних типи: статичні та динамічні параметри трафіку. Параметри першого типу характеризують ступінь заповненості дороги транспортними засобами, другого – інтенсивність руху транспортних засобів на дорозі.

Динамічні параметри трафіку. Розглянемо динамічні параметри трафіку. Класичним простим і найбільш поширеним методом для оцінки інтенсивності руху є показник AADT, що описує річний середньодобовий трафік на ділянці дороги [13]. Він вимірюється як кількість автомобілів, що проїхали ділянку дороги за рік, поділена на кількість днів у році. Цей

показник дозволяє оцінити ступінь завантаженості ділянки дороги за певний час та порівняти її з теоретичною максимальною пропускною здатністю, після чого можна зробити висновки щодо ефективності транспортних розв'язок у околицях цієї ділянки.

Проте показник AADT має свої обмеження. Він опускає таку важливу характеристику, як зміна інтенсивності за час виміру, оскільки оцінює середнє значення за весь рік. Однак сезонні, тижневі та добові цикли, які він не враховує, можуть істотно впливати на інтенсивність руху. Можливим рішенням є перехід від середньорічної оцінки до оцінки за час, що відповідає кожному циклу[14]. Таким чином, можна побачити залежність інтенсивності від сезону, місяця, дня тижня або часу доби, коли виконується вимірювання. Якщо скоротити інтервал вимірювань параметра до 1 хвилини або 1 години, отримаємо динамічну характеристику, яка може бути корисною для визначення ситуації на дорозі в режимі реального часу або максимально наближено до цього, що допоможе оперативно відреагувати на зміну ситуації на дорозі.

Однак, навіть із таким покращенням, показник AADT дозволяє лише оцінити кількість транспортних засобів, що пройшли ділянку дороги. З огляду на те, що кожна ділянка дороги може мати свій окремий швидкісний режим і враховуючи різноманітність транспортних засобів за розмірами, зробити висновки за абсолютними величинами буде доволі складно.

Іншим важливим параметром є швидкість руху транспортних засобів. Хоча швидкість і не дає повного уявлення про стан трафіку, вона є ключовим показником ефективності транспортного потоку, оскільки від неї безпосередньо залежить пропускна здатність дороги. Крім того, дані про швидкість дозволяють аналізувати розподіл швидкостей серед різних типів транспортних засобів, що може бути корисним при виявленні потенційних зон ризику дорожньо-транспортних пригод. Наприклад, можна виявити, що водії не зменшують швидкість перед потенційно небезпечним поворотом.

Окремо варто зазначити характеристику, як розподіл транспортних засобів за категоріями. Використовуючи методи комп'ютерного зору, можна класифікувати транспортні засоби за їхнім типом або іншими важливими ознаками для конкретного випадку. Це дозволяє деталізувати аналіз і надає можливість розробляти специфічні стратегії управління трафіком для різних категорій транспортних засобів, наприклад, приймати рішення щодо виділення окремої полоси руху під конкретний тип транспортних засобів.

Також варто звернути увагу на аналіз поведінки інших учасників дорожнього руху, таких як пішоходи або велосипедисти. Завдяки сучасним технологіям комп'ютерного зору, можна виявляти закономірності у їхньому русі та взаємодії з транспортними потоками. Наприклад, аналіз скупчення пішоходів на перехрестях може бути корисним для регулювання інтервалу роботи світлофорів та планування розвитку інфраструктури для пішоходів та велосипедистів.

Статичні параметри трафіку. Перейдемо до статичних характеристик дорожнього трафіку. На відміну від динамічних параметрів, статичні описують стан дороги у конкретний момент часу, зокрема ступінь завантаженості ділянки дороги транспортними засобами. Методи розрахунку цих характеристик здебільшого підпадають під категорію методів «ідентифікації заторів на дорозі». Для цього застосовуються різні підходи, включаючи визначення коефіцієнтів заповнюваності, що дозволяють оцінити рівень затору на дорозі.

Важливо зазначити, що для повного уявлення про стан дорожнього трафіку на ділянці дороги потрібно знати характеристики обох типів. Статична характеристика відповість на питання, наскільки багато транспортних засобів на ділянці дороги, а динамічна – на питання, наскільки швидко ці транспортні засоби проїжджають ділянку дороги. Тільки з комбінованим аналізом цих параметрів можна отримати точну картину про ситуацію на дорозі і відповідно прийняти обґрунтовані рішення щодо управління транспортним потоком.

Використання сучасних технологій для збору даних. Сучасні технології відіграють ключову роль у зборі та обробці даних про дорожній трафік.Одним із найбільш перспективних методів є використання сенсорних мереж та відеокамер, що дозволяють автоматично отримувати дані про інтенсивність руху, швидкість транспортних засобів, їх типи та інші важливі характеристики в режимі реального часу. Такі системи використовують камери спостереження з функцією комп'ютерного зору, що дозволяє точно визначати кількість і тип транспортних засобів, а також аналізувати їхню поведінку на дорозі.

Додатково, сучасні сенсорні мережі можуть включати різні типи датчиків, як-от магнітні, інфрачервоні або радарні, що дозволяють точно визначати кількість транспорту, що проїжджає через певну точку, а також фіксувати швидкість і час перебування кожного транспортного засобу на ділянці дороги. Ці дані можуть використовуватися для оцінки поточних умов на дорозі, прогнозування можливих заторів і навіть для виявлення порушень швидкісного режиму.

Інтелектуальні транспортні системи (ITS). Впровадження Інтелектуальних транспортних систем (ITS) є ще одним важливим кроком у вдосконаленні аналізу дорожнього трафіку. ITS включають використання різних інформаційних технологій для покращення управління транспортними потоками, підвищення безпеки на дорогах та зниження рівня забруднення навколишнього середовища. Ці системи поєднують різноманітні джерела інформації, такі як сенсори, камери спостереження, GPS-трекери в транспортних засобах та інші технології для моніторингу і аналізу трафіку в реальному часі.

Одним із основних завдань ITS є забезпечення оптимального руху транспорту на основі отриманих даних. Наприклад, за допомогою аналізу інтенсивності трафіку, швидкості руху і заторів на дорозі, система може автоматично коригувати режим роботи світлофорів, направляти водіїв по

менш завантажених маршрутах через мобільні додатки або інформаційні табло на дорогах, а також надавати рекомендації щодо об'їзду заторів.

Прогнозування та моделювання дорожнього трафіку. Для більш точного планування та управління дорожнім трафіком часто застосовуються методи прогнозування та моделювання. Одним з найбільш поширених підходів є використання математичних моделей, таких як моделі теорії черг,а також методи статистичного аналізу та машинного навчання для прогнозування змін в інтенсивності руху та можливих заторів.

Моделювання дорожнього трафіку дозволяє прогнозувати вплив різних факторів на ситуацію на дорозі, таких як погодні умови, святкові та вихідні дні, зміни в дорожній інфраструктурі або великі масові події. Наприклад, моделі можуть передбачати збільшення трафіку під час свят або великих спортивних подій і надавати рекомендації для управління потоками транспорту. Це дозволяє зменшити ризики виникнення заторів і підвищити загальну ефективність дорожнього руху.

Інтеграція з іншими системами. Для досягнення максимального ефекту, методи аналізу дорожнього трафіку повинні інтегруватися з іншими системами, такими як системи управління містом, екологічні монітори та навіть системи екстреного реагування. Наприклад, дані про затори та аварійні ситуації можуть автоматично передаватися до центрів управління містом, що дозволить оперативно змінити світлофорні цикли або направити екстрені служби на місце події.

Таким чином, використання інтегрованих підходів дозволяєстворювати більш ефективні та безпечні транспортні системи, які не лише покращують управління трафіком, але й знижують кількість дорожньо- транспортних пригод, зменшують затори та підвищують комфорт водіїв і пішоходів.

Методи аналізу дорожнього трафіку є важливою складовою частиною сучасних транспортних систем. Вони дозволяють ефективно управляти дорожнім потоком, зменшувати затори, підвищувати безпеку та знижувати

негативний вплив транспорту на навколишнє середовище. Застосування новітніх технологій, таких як комп'ютерне зору, Інтелектуальні транспортні системи, а також методи прогнозування та моделювання, значно покращують точність та оперативність збору та аналізу даних, що дозволяє прийматибільш обґрунтовані рішення щодо управління трафіком.

1.3 Методи виявлення транспортних засобів

Перед тим як перейти до вимірювання характеристик дорожнього трафіку, розглянемо методи, які дозволяють автоматизувати цей процес, а саме методи виявлення транспортних засобів. Якщо ми знаємо, що в певний час на ділянці дороги був присутній автомобіль, ми можемо використовуватиці дані для таких цілей, як підрахунок кількості автомобілів, що проїхали ділянку дороги, та активація фото- або відеофіксації цієї ділянки з подальшим аналізом, наприклад, для фіксації номерних знаків.

Традиційними методами для вирішення цієї задачі є використання різноманітних сенсорів, здебільшого радарів або вагових сенсорів. Вони базуються на фіксації зміни вихідних значень сенсора та дозволяють зафіксувати факт присутності транспортного засобу в області відповідальності сенсора на певний час. Як приклад, у статті [3] описано метод, що дозволяє це робити, а також підраховувати кількість автомобілів. В деяких випадках можливо навіть класифікувати транспортний засіб за вагою або розміром. Так, у статті [4] розглянуто метод використання переносного радару для підрахунку кількості автомобілів на ділянці дороги з можливістю їх класифікації та навіть оцінки їх швидкості.

Проте такі сенсори мають свої обмеження через вузьку спеціалізацію. Вони можуть не надавати достатньо точних або універсальних даних для всіх сценаріїв, таких як різні погодні умови або складні дорожні ситуації.

В контексті сенсорів, що фіксують наявність або відсутність транспортних засобів на дорозі, не можна не згадати відеокамери, які також є сенсорами. Базові методи засновані на фіксації зміни вихідних значень, що в

даному випадку є зміною кадрів відео або зміною пікселів на зображенні. У контексті відеокамер такими методами є методи фіксації руху. У статті [5] описано метод фіксації руху, що базується на зваженому відніманні зображень та векторі руху. У роботі [6] запропоновано підхід до виявлення таоцінки руху на основі візуальної уваги, що використовує дві різні техніки порогової обробки. Цей метод порівнюється з технікою оцінки руху Блека[7], яка базується на вимірюванні загального кута руху.

Значним поштовхом для розвитку виявлення об’єктів, в тому числі транспортних засобів, став розвиток машинного навчання та поява нейронних мереж. Окрім того, що стало можливим просто виявлятитранспортні засоби в полі зору сенсора, сучасні методи дозволяють точно визначати їхні положення, границі та навіть відносити їх до одного з класів. Для таких завдань стандартом стало використання нейронних мереж типу YOLO (You Only Look Once) [8].

Принцип роботи нейронних мереж типу YOLO зображений на Рис.1.1. Процес виявлення об'єктів відбувається в один етап, що дозволяє значно знизити час обробки зображення. Нейронна мережа за один прохід визначає координати та класи об'єктів на вхідному зображенні. Алгоритм розділяє зображення на сітку та призначає клас і координати об'єкта клітині з найвищою ймовірністю наявності об'єкта.

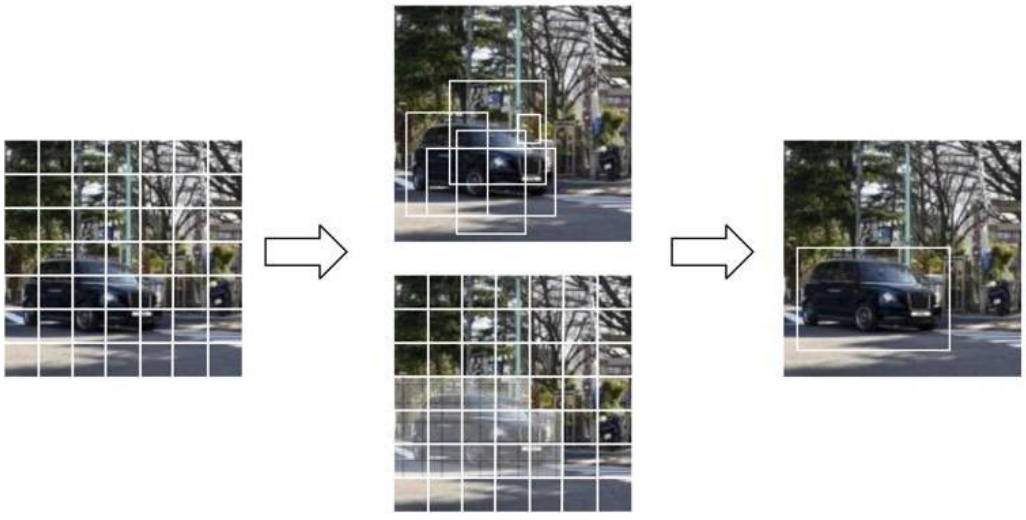

Рисунок 1.1 – Принцип роботи нейромереж типу YOLO [8]

Проте в ранніх версіях цієї нейронної мережі були проблеми з детекцією транспортних засобів, якщо вони накладалися один на одного. Часткове вирішення цієї проблеми було запропоновано у роботі [9] за допомогою використання придушення немаксимумів (Soft-NMS). Цяметодика дозволяє зменшити помилки в ситуаціях, коли об'єкти (наприклад, автомобілі) перекривають один одного, зберігаючи точність виявлення.

У сучасних версіях YOLO [10,11] ця проблема вирішена без компромісів зі швидкістю роботи нейронної мережі, що дозволяє точно та в реальному часі детектувати і підраховувати кількість автомобілів, що проходять ділянку дороги. Як продемонстровано у роботі [12] (Рис.1.2), сучасні версії YOLO здатні виконувати це завдання з високою точністю.Крім того, у сучасних версіях нейронної мережі вже не виникає складнощів з додаванням класифікації транспортних засобів, що дозволяє розширити її функціонал, надаючи додаткову інформацію про категорії транспортних засобів, що проходять через ділянку дороги.

Рисунок 1.2 – Результат роботи нейромережі YOLOv7 [12]

Таким чином, сучасні методи виявлення транспортних засобів, засновані на нейронних мережах, значно покращили точність і швидкість виявлення об'єктів на дорозі, що робить їх ефективними для використання в реальних умовах дорожнього руху, забезпечуючи автоматичний підрахунок і класифікацію транспортних засобів у реальному часі.

1.4 Методи визначення динамічних характеристик дорожнього трафіку

До основних динамічних параметрів трафіку можна віднести інтенсивність руху, що описує кількість автомобілів, які проїхали певну ділянку дороги за певний проміжок часу, а також середню швидкість цих автомобілів. Розглянемо існуючі методи, які дозволяють визначити ці параметри або їх похідні.

Один із основних методів оцінки інтенсивності руху – це використання показника AADT (Annual Average Daily Traffic), який відображає середньодобовий трафік протягом року на певній ділянці дороги. Його розраховують, поділивши загальну кількість автомобілів, що проїхали цю ділянку за рік, на кількість днів у році. Цей показник дозволяє оцінити рівень навантаження на дорогу та порівняти його з максимально можливим обсягом руху. Водночас важливо враховувати зміни в інтенсивності руху у різні часові періоди, такі як сезонні, тижневі та добові цикли. Оскільки цей показник дає середнє значення трафіку, він не завжди відображає пікові навантаження на дорогу в конкретні моменти часу, що може бути критичним при плануванні інфраструктури та організації руху на конкретних ділянках доріг.

Ранні рішення щодо визначення швидкості транспортних засобів та інтенсивності руху з’явилися досить давно. Проте ці рішення вимагали розробки та впровадження спеціалізованого обладнання з детекторами [13,14]. Це дозволило точно вимірювати швидкість транспортних засобів,однак були проблеми з тим, що рішення були апаратними, а не програмними.

Тобто для того, щоб впровадити інший метод, необхідно було переробляти всю систему спостереження. Такі апаратні методи мали обмеження щодо швидкості їхньої адаптації до нових умов, таких як зміни інтенсивності трафіку або вимоги до точності вимірювань у складних умовах.

У роботі [15] було запропоновано метод оцінки швидкості транспортних засобів за допомогою даних з відеокамер. Однак цей метод вимагає, щоб камера була встановлена перпендикулярно до дороги. Тому необхідно виділяти окрему камеру для розрахунку швидкості, що ускладнює стандартизацію (Рис.1.3). Для точних вимірювань швидкості транспортних засобів за допомогою відеокамер також необхідно мати високу роздільну здатність камери та спеціальні алгоритми обробки зображень, що додає до вартості таких систем. Проблема перпендикулярного розташування камери також обмежує можливість її ефективного використання в умовах, коли дорожня інфраструктура не дозволяє встановити камеру в такому положенні.

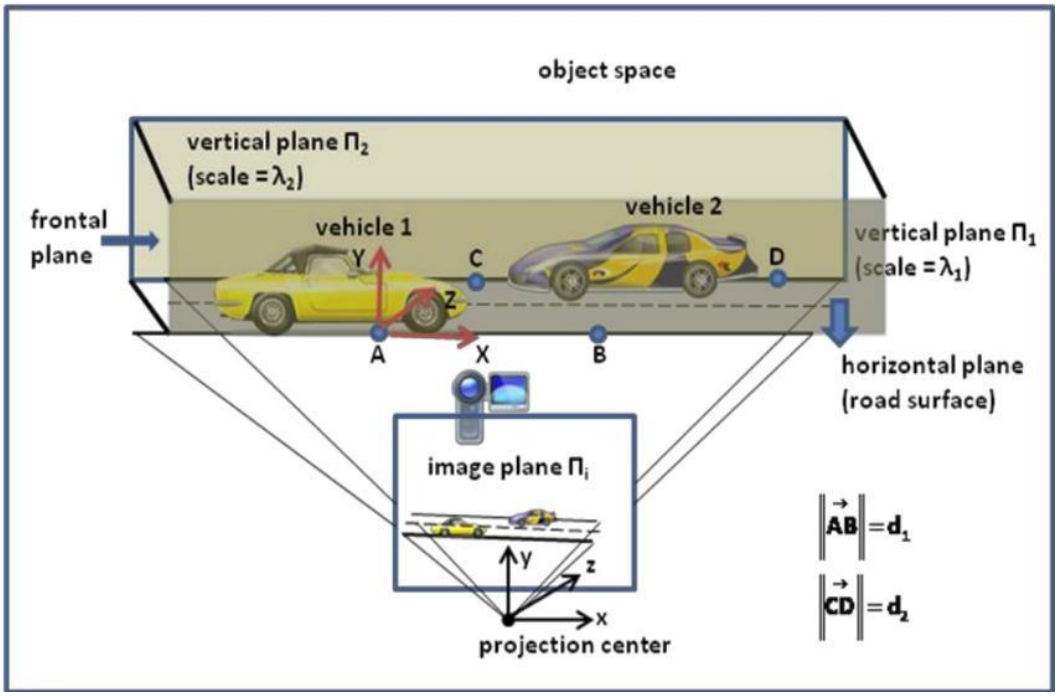

Рисунок 1.3 – Оцінювання швидкості транспортних засобів за допомогою виду збоку (https://example.com) [15]

Метод, запропонований у статті *Analytical Modeling for a Video-Based Vehicle Speed Measurement Framework* [16], частково вирішує цю проблему, адже камера тепер повинна бути розташована над дорогою (Рис.1.4). Це дозволяє виконувати аналіз даних з більшої частини вже встановлених камер відеоспостереження. Встановлення камер над дорогою дозволяє зменшити вплив таких факторів, як зміна умов освітлення, кути нахилу дороги та інші перешкоди, що можуть спотворювати результати вимірювань. Цей метод дозволяє зібрати більше даних і забезпечити більш точні результати, оскільки камери можуть охоплювати більшу частину дороги, зменшуючи зону "мертвих" зон, де неможливо здійснити точне вимірювання.

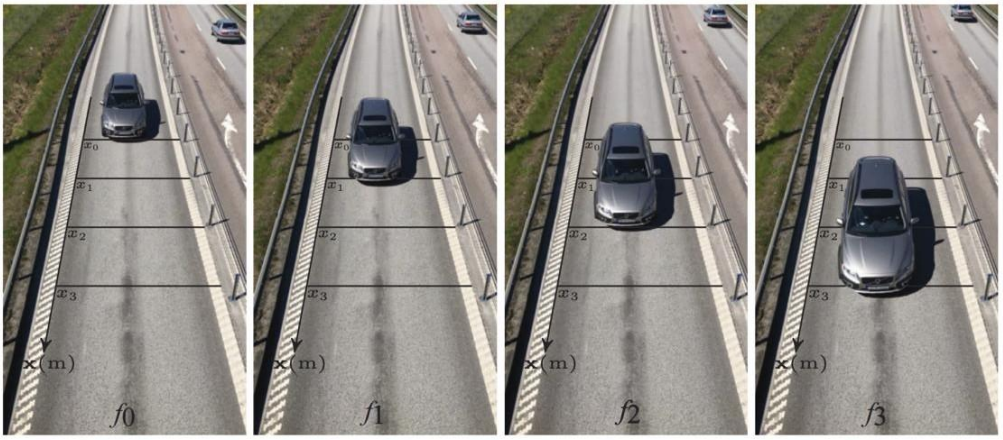

Рисунок 1.4 – Оцінювання швидкості транспортних засобів за допомогою виду зверху (https://example.com) [16]

Повертаючись до методів машинного навчання, у статті [17] описана альтернатива для виявлення, трекінгу та підрахунку кількості транспортних засобів у трафіку за допомогою сегментації зображень. Однак цей методможе стикатися з труднощами, коли на досліджуваній ділянці дороги автомобілі часто зупиняються або перекривають один одного, наприклад, поблизу світлофорів або перехресть. У таких випадках цей метод підрахунку транспортних засобів буде видавати неточні оцінки стану трафіку на дорозі. Окрім цього, методи сегментації зображень, як правило, потребують

потужних обчислювальних ресурсів для обробки великих масивів відеоданих у реальному часі, що може бути додатковим бар'єром для їх широкого застосування.

У статті *Real-time Traffic Analysis Using Deep Learning Techniques and UAV-Based Video* [18] пропонується використовувати відеодані, зібрані за допомогою безпілотників (UAV), у поєднанні з методами глибинногонавчання для виявлення автомобілів на відеозаписах та визначення їх швидкостей. Однак через використання безпілотників автори накладають на себе певні обмеження. По-перше, залежність цього методу від погодних умовта можливі труднощі з його адаптацією до різних міських сценаріїв та швидко змінюваних умов можуть знизити його загальну ефективність у практичному застосуванні, оскільки дрон не може працювати цілодобово. По-друге, цей аналіз не є аналізом у реальному часі, тому він не надає актуальних даних для швидкої реакції на різні ситуації на дорозі. Однак цей підхід дозволяє зробити систему аналізу трафіку адаптивною до поточних потреб, оскільки не вимагає будівництва інфраструктури та встановлення камер для аналізу ділянки дороги. В цілому, хоча практична ефективність підходу, запропонованого в статті, значно обмежена використанням безпілотників для отримання відеоматеріалів, методи, що використовуються для аналізу, досить легко адаптувати для системи аналізу в реальному часі з використанням стаціонарних камер.

Загалом, сучасні методи аналізу дорожнього трафіку відзначаються великою різноманітністю підходів і технологій, що дозволяють отримати точні дані щодо стану руху на дорогах. Однак ефективність кожного з методів значною мірою залежить від конкретних умов і вимог до системи спостереження, а також від технічних можливостей застосовуваних пристроїв.

1.5 Методи визначення статичних характеристик дорожнього трафіку

Статичні характеристики дорожнього трафіку дозволяють оцінювати стан транспортних потоків на конкретній ділянці дороги в певний момент часу, що є важливим для моніторингу та управління дорожнім рухом. Визначення цих характеристик дає змогу аналізувати завантаженість ділянки, виявляти потенційні місця для заторів і прогнозувати трафік на основі отриманих даних. Зазвичай вони оцінюються у вигляді кількості автомобілів, які перебувають на дорозі, або через визначення відношення площі, яку займають автомобілі, до загальної площі спостережуваної ділянки.

Один із основних методів збору таких даних – використання GPS-трекерів. У статті [19] описано використання GPS-трекерів для моніторингу трафіку на різних ділянках доріг. Цей метод дозволяє отримувати точні дані про рух автомобілів в реальному часі, однак існують певні обмеження щодо охоплення території. Для того, щоб забезпечити повне охоплення трафіку на всіх ділянках дороги, необхідно мати велику кількість активних GPS- трекерів, що є складним завданням, особливо на великих або багатосмугових дорогах. Крім того, для отримання коректних результатів важливо, щоб транспортні засоби були оснащені відповідними пристроями, а їх сигнал був достатньо сильним і стабільним.

Іншим методом є використання відеоаналізу для визначення завантаженості дороги. Наприклад, у статті *Smart traffic lights switching and traffic density calculation using video processing* [20] пропонується система, яка динамічно регулює інтервали перемикання світлофорів, оцінюючи завантаженість дороги. Для цього система порівнює поточний кадр з еталонним, на якому дорога порожня, і на основі цього порівняння робить висновки щодо кількості автомобілів на ділянці. Такий метод дозволяє здійснювати регулювання трафіку в реальному часі, але має свої обмеження. Важливими проблемами є вплив зміни освітлення, погодних умов та часу

доби, які можуть впливати на якість зображення і, відповідно, на точність аналізу.

Метод, запропонований у роботі *Traffic Lane Congestion Ratio Evaluation by Video Data* [21], використовує показник завантаженості *Traffic Lane Congestion Ratio (TLCR)*, який дозволяє оцінити щільність трафіку на смузі дороги, спрощуючи відеоаналіз. TLCR обчислюється шляхом визначення меж досліджуваної смуги руху та сегментації відеозображення за допомогою нейронної мережі U-Net [22] для виявлення транспортних засобів. Після цього обчислюється співвідношення пікселів, що позначають транспортні засоби, до загального числа пікселів у межах смуги. Результат цього аналізу знаходиться в діапазоні від 0 до 1, де значення, близьке до 1, вказує на високу завантаженість дороги.

Цей метод дозволяє значно зекономити обчислювальні ресурси, оскільки для оцінки завантаженості достатньо одного кадру. Крім того, його можна використовувати для регулярного моніторингу ситуації на дорозі, розраховуючи показник TLCR для кожного кадру з певним інтервалом. Однак є певні складнощі, пов'язані з перспективою на зображеннях, що може спотворювати результати при віддаленому або наближеному розташуванні камери. Також важливим фактором є можливість перекриття автомобілями, що рухаються по сусідніх смугах, що може призвести до заниження оцінки завантаженості.

Метод TLCR має також певні обмеження, пов'язані з необхідністю тренування нейронної мережі для виділення транспортних засобів на зображеннях. Це вимагає великих обсягів даних для навчання та може бути ускладнено через різноманітність типів транспортних засобів, їх кольорів і розмірів, що створює труднощі при автоматичній класифікації об'єктів.

Висновок: кожен з методів визначення статичних характеристик дорожнього трафіку має свої переваги та обмеження. Використання GPS-трекерів дозволяє отримувати точні дані про рух транспортних засобів, однак вимагає великої кількості пристроїв і може бути складним у реалізації на

великих ділянках. Відеоаналіз, зокрема методи на основі нейронних мереж, дозволяють оцінювати завантаженість дороги без необхідності збору даних від усіх транспортних засобів, проте мають обмеження, пов'язані з якістю зображення та складністю навчання моделей. Кожен з цих підходів є важливим для моніторингу та управління дорожнім рухом, особливо у поєднанні з іншими методами для забезпечення більш точної та ефективної оцінки трафіку.

1.6 Постановка задачі та висновки до розділу

Аналіз методів визначення статичних характеристик дорожнього трафіку засвідчує їх важливість для ефективного управління дорожнім рухом, виявлення потенційних заторів та планування інфраструктури. Різні підходи до оцінки завантаженості дороги мають свої сильні і слабкі сторони, що вимагає комплексного підходу для отримання точних результатів. Використання GPS-трекерів надає точні дані про місцезнаходження та швидкість транспортних засобів, але потребує значних витрат на підтримку великої кількості пристроїв та складності в інтеграції на великих ділянках доріг. З іншого боку, відеоаналіз є доступним і ефективним методом, особливо в умовах високої щільності трафіку, однак його точність можезнижуватися в умовах змін у освітленні, погодних умовах або перекриття смуг транспортними засобами.

Метод TLCR, який розраховує завантаженість смуги руху через сегментацію відео, дозволяє оперативно оцінювати рівень завантаженості дороги, що є важливим для адаптивних систем управління дорожнім рухом. Однак, цей підхід вимагає вдосконалення для роботи в умовах перспективи та перекриття транспортними засобами. У зв'язку з цим важливим є подальше вдосконалення технологій та алгоритмів, які зможуть забезпечити більш точні й надійні результати, знижуючи обчислювальні витрати та адаптуючись до різноманітних умов.

Постановка задач дослідження:

1. Розширення можливостей GPS-трекерів для моніторингу трафіку. Необхідно дослідити способи збільшення ефективності використання GPS-трекерів для широких і багатосмугових ділянок, що дозволить покращити точність визначення інтенсивності трафіку. Важливо визначити можливості інтеграції цього методу з іншими технологіями для покриття великих територій.

2. Удосконалення методів відеоаналізу для розпізнавання трафіку в умовах змінюваних умов. Потрібно створити більш адаптивні алгоритми відеоаналізу, що будуть стійкими до змін освітлення, погодних умов та інших факторів, що можуть вплинути на якість відео. Це включає в себе використання новітніх технологій машинного навчання для підвищення точності та швидкості обробки відеоданих.

3. Оптимізація обчислювальних ресурсів при використанні методів відеоаналізу. Для забезпечення ефективності використання відеоаналізу в реальному часі важливо розробити алгоритми, які зможуть знижувати обчислювальні витрати, зберігаючи необхідну точність оцінок. Це включає в себе вдосконалення технік сегментації зображень та їх адаптацію дореальних умов дорожнього руху.

4. Покращення навчання нейронних мереж для аналізу дорожнього трафіку. Для підвищення ефективності нейронних мереж в задачах розпізнавання трафіку важливо розробити більш потужні моделі для обробки складних та різноманітних умов. Це потребує збору більш репрезентативних наборів даних та застосування нових архітектур нейронних мереж, здатних до точного розпізнавання транспортних засобів у реальному часі.

5. Інтеграція різних методів для комплексного моніторингу трафіку. Для підвищення точності оцінки завантаженості доріг пропонується розробка систем, що поєднують GPS-трекери, відеоаналіз та інші технології в єдину платформу для комплексного моніторингу та прогнозування трафіку. Це дозволить забезпечити більш точні та своєчасні дані для управління дорожнім рухом і планування інфраструктури.

2 РОЗРОБКА МЕТОДІВ ОЦІНКИ ЩІЛЬНОСТІ ТА ІНТЕНСИВНОСТІ ТРАФІКУ

2.1 Модифікований показник TLCR

Як вже зазначено, у дослідженні [21] було введено показник TLCR (Traffic Lane Congestion Ratio), який визначає ступінь заповненості смуги руху. У роботі [1] було проаналізовано метод обчислення цього показника та ситуації, в яких він може давати неточні результати. Тому було запропоновано ввести модифікований показник — MTLCR. Розглянемо ці ситуації на прикладі кадру з відеокамери, зображеного на Рис. 2.1.

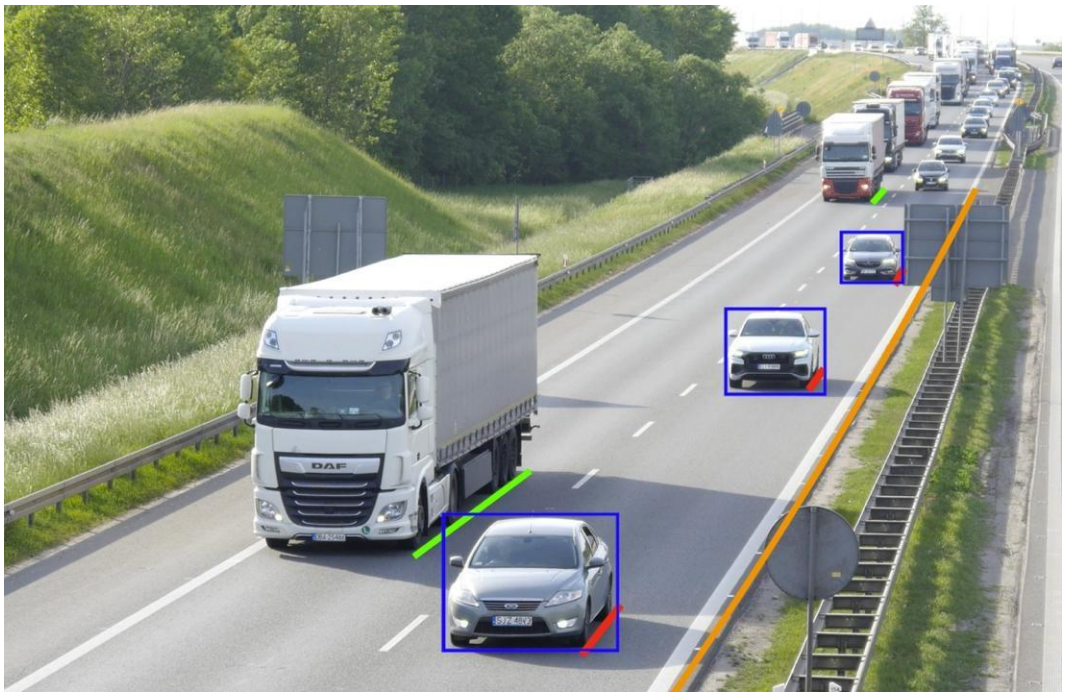

Рисунок 2.1 – Кадр з камери відеоспостереження біля міста Вроцлав, Польща[23].

Спочатку розглянемо смугу дороги, по якій рухаються вантажні автомобілі. Оскільки кадр зроблений на автомагістралі в Польщі, і довжина тягача з причепом є нормованою в Європі, можна припустити, що всі вантажні автомобілі на цій смузі мають приблизно однакову довжину і, отже,повинні робити приблизно однаковий внесок у завантаженість смуги. Проте

через вплив перспективи внесок транспортних засобів, що знаходяться ближче до камери (або їх відсутності), значно більший, ніж у транспортних засобів, що знаходяться далеко. Вплив перспективи на розміри вантажних автомобілів показано зеленими лініями на рисунку. Аналогічно, вплив перспективи на смугу з легковими автомобілями показано червонимилініями. Отже, якщо обчислити показник TLCR для цієї смуги, то буде видно, що внесок п'яти вантажних автомобілів, що знаходяться на задньому плані, дорівнює приблизно половині внеску першого вантажного автомобіля, при тому що в реальності всі вантажні автомобілі мають приблизно однаковий розмір. В оригінальній статті ця проблема частково вирішується тим, що для розрахунку показника відокремлюється невеликий відрізок дороги, близький до камери відеоспостереження, де може розташуватися максимум декілька легкових автомобілів, проте вплив перспективи навіть на такій ділянці буде помітний. Крім того, таке обмеження досліджуваної ділянки впливає на правильність оцінки TLCR через нерівномірність транспортного потоку.

Тепер розглянемо іншу потенційну проблему. Звернемо увагу на автомобілі, обведені синіми прямокутниками на Рис.2.1. Якщо використовувати метод, за яким TLCR обчислюється як відношення кількості пікселів, що припадають на автомобілі, до кількості пікселів у прямокутнику, що позначає смугу, то в такому випадку верхні частини легкових автомобілів будуть впливати на значення показника завантаженості для смуги з вантажними автомобілями.

Також фактичний розрахунок показника як відношення зайнятої площі смуги руху до загальної площі має ще два негативні впливи на результуючий показник. По-перше, на практиці значення показника майже ніколи не перевищує 0,6, навіть у ситуаціях, коли транспортні засоби стоять дуже щільно один до одного, через те, що вони не займають усю горизонтальну площу смуги руху. Середня довжина транспортних засобів приблизно 2 метри, а ширина смуги руху — 3,5 метра. В результаті ми маємо проміжок можливих значень показника, який фактично не використовується. По-друге,

вже згадана ширина та додатково висота транспортного засобу також впливає на розрахунок значення показника TLCR. Так, якщо на одній й тій самій ділянці дороги будуть присутні два автомобілі з однаковою довжиною, проте з різними шириною та висотою, то для кадру, на якому зображений автомобіль з більшими габаритами, значення показника TLCR буде більшим, хоча теоретично в обох випадках автомобілі повинні мати однаковий вплив на щільність трафіку.

Для вирішення проблем, описаних вище, було запропоновано модифікацію алгоритму обчислення показника TLCR. Цей алгоритм передбачає кілька ключових етапів, кожен з яких сприяє точному та ефективному визначенню завантаженості смуги руху:

1. Визначення меж досліджуваної смуги руху як чотирикутник. Першим етапом є визначення меж досліджуваної смуги руху у вигляді чотирикутника, що дозволяє точно обмежити область аналізу та зосередити увагу на конкретній ділянці дороги.

2. Сегментація зображення за допомогою нейромережі U-Net для сегментації дорожнього полотна. Далі виконується сегментація зображення за допомогою натренованої нейромережі U-Net. Ця мережа спеціалізується на сегментації зображень і дозволяє виокремити дорожнє полотно на зображенні. Вона розпізнає пікселі, що належать дорозі, і виділяє їх окремо від пікселів, що належать іншим об'єктам, таким як транспортні засоби.

3. Перспективне перетворення зображення для вирівнювання геометрії. На цьому етапі використовується перспективне перетворення за допомогою бібліотеки OpenCV. Перспективне перетворення дозволяє "вирівняти" зображення ділянки дороги, переводячи його в прямокутний формат. Це забезпечує спрощене та чітке подання смуги руху, на якому видно, де саме знаходяться транспортні засоби і де — дорога.

4. Згортання та порогова операція. Після виконання перспективного перетворення зображення проводиться операція згортання. Цей етап включає прохід по прямокутнику вздовж бічного краю дороги. Для кожного перерізу

дороги в прямокутнику виконується порогова операція: якщо кількість пікселів, які не належать дорожньому полотну, перевищує заданий поріг, то результат операції буде дорівнювати 1, в іншому випадку — 0. Це дозволяє визначити наявність або відсутність завантаженості в конкретному перерізі дороги.

5. Обчислення модифікованого TLCR. Розрахунок модифікованого показника TLCR проводиться як співвідношення кількості одиниць у масиві до загальної кількості елементів в масиві. Це дає точне значення завантаженості смуги руху, відображаючи, наскільки заповнена дорога транспортними засобами.

Пропонована модифікація алгоритму спрямована на вирішення кількох важливих проблем:

- Сегментація різних типів транспортних засобів: Як зазначено, для точної сегментації різних типів транспортних засобів необхідно мати велику кількість тренувальних даних. Однак це потребує значних ресурсів і часу для навчання моделі. Тому для вирішення цієї проблеми пропонується сегментація саме дорожнього полотна за допомогою нейромережі U-Net. Це значно спрощує задачу, оскільки сегментація дорожнього полотна є набагато менш складною, ніж сегментація різних типів транспортних засобів.

- Адаптивність до нових типів транспортних засобів: Використання нейромережі для сегментації тільки дорожнього полотна означає, що не потрібно перетреновувати модель кожного разу, коли з'являється новий тип транспортного засобу. Це робить систему більш універсальною та адаптивною.

- Присутність об'єктів, не пов'язаних з транспортними засобами: Іноді на дорозі можуть бути не тільки транспортні засоби, а й інші об'єкти, наприклад, будівельні матеріали чи дорожні роботи. Враховуючи це, модифікований алгоритм дозволяє враховувати такі ситуації та коректно відображати завантаженість смуги руху, навіть коли на дорозі немає транспорту.

Запропонована модифікація алгоритму розв'язує кілька важливих проблем, таких як складність сегментації різних типів транспортних засобів і необхідність адаптації до нових умов, що з'являються на дорозі. Завдяки використанню сегментації дорожнього полотна та перспективному перетворенню зображень, алгоритм забезпечує точну оцінку завантаженості смуги руху в різних умовах.

Для наочної демонстрації того, як це впливає на зображення, проведемо цю операцію на вихідному зображенні відразу для двох смуг руху (рис. 2.2). Після того, як виконано перетворення з урахуванням перспективи, розміри усіх об’єктів, що знаходяться у площині дороги, виправилися. Це видно по зелених лініях вздовж вантажних автомобілів, а також по однаковому інтервалу розмітки переривчастої лінії між смугами.

Для кращого розуміння, як перетворення перспективи впливає на зображення, розглянемо обробку зображення двох смуг руху (див. рис. 2.2). Після того, як було виконано перетворення з урахуванням перспективи, розміри всіх об'єктів, які знаходяться на площині дороги, були коректно відображені. Це чітко видно на прикладі зелених ліній, що позначають вантажівки, а також завдяки однаковому інтервалу переривчастої лінії між смугами руху.

Рисунок 2.2 – Зображення ділянки дороги після виконання перспективного перетворення.

Однак цей метод має одну суттєву проблему: неможливо позбутися вертикальної складової проекції автомобілів на зображенні. Коли перспектива коригується, автомобілі, у яких вертикальна складова значно переважає горизонтальну, особливо на більших відстанях, виглядають сильно розтягнутими по вертикалі. Це може серйозно вплинути на точність розрахунків завантаженості смуги. Як видно на Рис.2.2, після коригування перспективи перший фрагмент дороги до другого легкового автомобіля зображений досить коректно. Однак на більш віддалених ділянках дороги вертикальна складова значно переважає горизонтальну, і автомобілі на цих ділянках розтягуються по вертикалі. Це стає особливо помітним при порівнянні з автомобілями, що знаходяться ближче до камери. Щоб вирішити цю проблему, можна застосувати два підходи:

1) Аналізувати лише ті ділянки дороги, де автомобілі знаходяться на відносно невеликій відстані від камери.

2) Розробити методику для видалення вертикальної складової проекції транспортних засобів ще до етапу сегментації зображення.

Рішення цієї проблеми продовжується в кроках 4 та 5, де аналізуються частини зображень, що потрапляють на сусідні смуги. Після третього кроку ми отримуємо прямокутну проєкцію дороги, де вертикальну складову можна співвіднести з довжиною дороги, а горизонтальну – з її шириною. Далі розглядається кожен окремий рядок пікселів зображення досліджуваної смуги. Після сегментації зображення ми можемо побачити, які пікселі належать до транспортних засобів, а які – до дороги. Важливо зазначити, що ширина автомобіля не повинна впливати на визначення завантаженості смуги, оскільки кожен рядок пікселів є або заповненим транспортним засобом, або порожнім.

Тепер розглянемо, як зміниться розрахунок показника завантаженості смуги. Спочатку для кожної ділянки довжини смуги визначається, чи є на ній транспортний засіб. Для кожного горизонтального рядка пікселів по всій

ширині дороги підраховуються пікселі, що належать транспортному засобу. Якщо співвідношення білих пікселів до загальної кількості перевищує заданий поріг, то на цій ділянці дороги вважається наявним транспортний засіб. В результаті, ми отримуємо одновимірний масив, довжина якого дорівнює кількості пікселів, що відображають досліджувану смугу, де кожен елемент масиву вказує на наявність або відсутність транспортного засобу на певному перерізі дороги.

Таким чином, змінений показник завантаженості смуги (MTLCR) розраховується за формулою:

$$MTLCR = \frac{N_v}{N}$$

де N – загальна кількість елементів масиву, Nv – кількість елементів, що відповідають наявності транспортного засобу на даній ділянці.

Для розрахунку MTLCR для смуги з легковими автомобілями використовується послідовність перетворень, яка ілюструється на Рис.2.3. Спочатку виділяється прямокутний фрагмент зображення, що підлягає обробці, позначений синіми межами на рисунку. Потім цей фрагмент обробляється нейронною мережею U-Net, що проводить сегментацію пікселів, відокремлюючи дорожнє полотно від інших об'єктів. Після цього здійснюється перспективне перетворення виділеної ділянки зображення, щоб правильно відобразити його на площині. Завершальний етап включаєобробку кожного горизонтального рядка пікселів для отримання результату увигляді масиву.

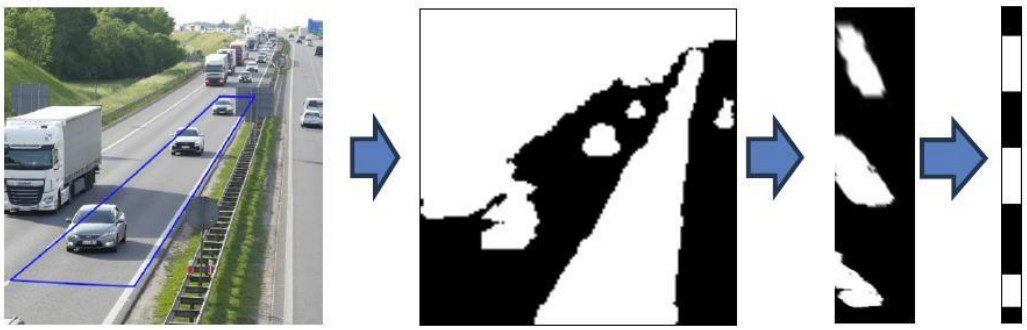

Рисунок 2.3 – Алгоритм перетворення зображень для обчислення показника MTLCR.

В цьому прикладі використовувалося порогове значення 0,25 від загальної кількості пікселів. Для наочності, масив був відображений у вигляді вертикальної смуги. Після цього, використовуючи формулу (2.1), розраховуємо показник MTLCR. У результаті для даного зображення MTLCR = 0,509. Це значення показує відсоток часу, коли смуга була зайнята транспортними засобами на основі сегментації та коригування перспективи.

2.2 Показник інтенсивності дорожнього трафіку Traffic Lane Intensity Ratio (TLIR)

Показник інтенсивності дорожнього трафіку (Traffic Lane Intensity Ratio, TLIR) є важливим параметром для оцінки стану руху на дорозі. Він дозволяє не лише визначити, скільки автомобілів проходять через певну ділянку за одиницю часу, але й порівняти цю величину з максимально допустимою інтенсивністю руху для конкретної дороги.

Як зазначено в описі, існуючі методи оцінки інтенсивності руху часто базуються на кількості транспортних засобів, які проходять через ділянку дороги за певний час. Це дає загальне уявлення про інтенсивність трафіку, проте ці методи мають обмеження:

1. Абсолютне значення інтенсивності не дозволяє оцінити, наскільки ефективно використовуються дорожні ресурси, адже без урахування параметрів дороги та швидкісного режиму неможливо зрозуміти, чи є цей рівень інтенсивності оптимальним для конкретної ділянки.

2. Коливання інтенсивності можуть бути значними на коротких проміжках часу, коли ситуація на дорозі змінюється кілька разів протягом однієї години чи доби. Тому для точнішої оцінки інтенсивності необхідно враховувати короткострокові зміни в русі.

Щоб усунути ці обмеження, пропонується новий універсальний показник інтенсивності, TLIR, який відображатиме фактичну інтенсивність трафіку у відношенні до максимально допустимої інтенсивності для конкретної ділянки дороги. Це дозволяє більш точно оцінювати стан трафіку в реальному часі.

Формула для розрахунку TLIR виглядає наступним чином:

$$TLIR = \frac{I}{I_{\max}}$$

де:

I — фактична інтенсивність руху (кількість транспортних засобів, що проходять через ділянку за одиницю часу),

$I_{\max}$ — максимально допустима інтенсивність руху для цієї ділянки.

Інтенсивність руху визначається як кількість транспортних засобів, що проїжджають через спостережувану ділянку дороги за одиницю часу.Оскільки транспортні засоби можуть бути різного типу (легкові автомобілі, автобуси, вантажівки тощо), для стандартизації часто використовують умовного автомобіля. Це означає, що різні типи транспортних засобів конвертуються в умовні одиниці, щоб забезпечити узгодженість оцінки інтенсивності руху. Тому інтенсивність вимірюється в умовних автомобілях за одиницю часу.

Такий підхід дозволяє отримати більш точну картину руху на дорозі, а також визначити, чи відповідає інтенсивність руху оптимальним значенням для забезпечення безпеки та ефективності дорожнього потоку.

Для оцінки фактичної інтенсивності руху без необхідності відслідковувати кожен автомобіль окремо або вимірювати його довжину в умовних автомобілях, можна скористатися показником MTLCR (Maximum Traffic Lane Coverage Ratio), який відображає відсоток смуги, зайнятий транспортними засобами. Це дає можливість оцінити, яка частина дороги

використовується для руху транспортних засобів, не вдаючись до складних обчислень для кожного окремого автомобіля.

Кроки для оцінки інтенсивності:

1. Визначення довжини ділянки, зайнятої транспортними засобами: Після обчислення значення MTLCR можна визначити довжину ділянки дороги, яку займають транспортні засоби. Це можна виразити за допомогою наступної формули:

$$L = \frac{MTLCR \cdot L_{\text{road}}}{100}$$

де:

L — довжина ділянки, яку займають транспортні засоби (у метрах),

MTLCR — відсоток смуги, зайнятий транспортними засобами,

L_{road} — довжина спостережуваної ділянки дороги (у метрах).

Далі цю довжину можна перевести в умовні автомобілі (Na), використовуючи довжину умовного автомобіля (La), що дозволить здійснити стандартизовану оцінку інтенсивності.

$$Na = \frac{L}{L_a}$$

де:

Na — кількість умовних автомобілів,

L_a — довжина умовного автомобіля (у метрах).

2. Визначення середньої швидкості руху: Для того, щоб оцінити фактичну інтенсивність руху на смузі, потрібно також врахувати швидкість потоку автомобілів. Середню швидкість руху можна визначити за допомогою нейромережі YOLOv8, яка виявляє транспортні засоби на кадрах відео та відстежує їх рух.

Протягом кількох кадрів відбувається вимірювання швидкості кожного автомобіля, а середня швидкість V_a обчислюється на основі цих даних:

$$V_a = \frac{1}{n} \sum_{i=1}^{n} V_i$$

де:

V_a — середня швидкість смуги,

V_i — швидкість кожного транспортного засобу, що спостерігається на кадрах,

n — кількість автомобілів, що спостерігаються в рамках аналізу.

Важливо зазначити, що заміри швидкості повинні здійснюватися протягом короткого проміжку часу, щоб мінімізувати вплив змін на трафік. Це особливо критично в умовах відсутності трафіку, коли середня швидкість руху може швидко змінюватися і наближатися до нуля.

Цей підхід дозволяє зменшити складність обчислень, не вдаючись до точного відстеження кожного транспортного засобу, але водночас зберігає високу точність у вимірюванні інтенсивності руху і швидкості потоку. Поєднання MTLCR, довжини смуги, кількості умовних автомобілів та швидкості руху дає змогу створити більш гнучкий і швидкий метод дляоцінки інтенсивності дорожнього трафіку в реальному часі.

З урахуванням отриманих даних, ми можемо підсумувати обчислення фактичної інтенсивності руху та визначити максимальну інтенсивність.

Кроки для обчислення фактичної інтенсивності руху (TLIR):

1. Визначення часу T:

Як зазначено, час T — це час, за який умовний автомобіль (зі швидкістю V_a проїде всю спостережувану ділянку дороги. Оскільки рух транспортних засобів може бути нерівномірним, для обчислення часу можна скористатися середньою швидкістю потоку:

$$T = \frac{L_{\text{road}}}{V_a}$$

де:

T — час, за який умовний автомобіль покидає спостережувану ділянку дороги,

L_{road} — довжина спостережуваної ділянки дороги,

V_a — середня швидкість руху на смузі.

2. Обчислення фактичної інтенсивності руху:

Фактичну інтенсивність руху I можна визначити, скориставшись значенням довжини ділянки, яку займають транспортні засоби (виражену в умовних автомобілях Na та часом T, необхідним для проходу цієї ділянки:

$$I = \frac{Na}{T}$$

Це дасть фактичну інтенсивність руху на спостережуваній ділянці дороги.

Кроки для обчислення максимальної інтенсивності руху:

1. Максимальна швидкість на смузі:

Для розрахунку максимальної інтенсивності необхідно використати максимально допустиму швидкість , $V_{\max}$ що є найбільше можливою швидкістю для цієї дороги за її умовами.

2. Обчислення максимальної інтенсивності:

Максимальна інтенсивність $I_{\max}$ буде визначена як кількість транспортних засобів, що можуть пройти через спостережувану ділянку дороги за одиницю часу при максимально можливій швидкості для смуги:

$$I_{\max} = \frac{L_{\text{road}}}{V_{\max} \cdot \Delta t}$$

де:

$I_{\max}$ — максимальна інтенсивність,

L_{road} — довжина ділянки дороги,

$V_{\max}$ — максимально допустима швидкість для цієї смуги,

Δt — максимальний допустимий інтервал між транспортними засобами.

Показник інтенсивності дорожнього руху (TLIR):

Тепер, маючи значення фактичної інтенсивності I та максимально можливої інтенсивності $I_{\max}$, ми можемо визначити показник Traffic Lane Intensity Ratio (TLIR):

$$TLIR = \frac{I}{I_{\max}}$$

Цей показник дає відношення фактичної інтенсивності до максимально можливого значення, що дозволяє оцінити, наскільки ефективно використовуються дорожні смуги. Важливо, що значення TLIR варіюється від 0 до 1:

- 0 — трафік відсутній (ні один автомобіль не рухається),

- 1 — трафік на максимальному рівні (дорога повністю заповнена транспортними засобами з максимально допустимою швидкістю).

Таким чином, ми маємо повний метод для оцінки інтенсивності дорожнього руху, враховуючи різні динамічні фактори, такі як швидкість потоку і рівень завантаженості дороги.

2.3 Система показників для визначення поточного стану трафіку на ділянці дороги

У цьому розділі була розглянута система показників, що дозволяє оцінити поточний стан трафіку на ділянці дороги. За допомогою двох основних показників — MTLCR (Median Traffic Lane Coverage Ratio) та TLIR (Traffic Lane Intensity Ratio) — можна визначити, як змінюєтьсязавантаженість дороги і інтенсивність руху в реальному часі. Ці показники є відносними, що дає можливість робити висновки про стан трафіку без

необхідності знати конкретні характеристики дороги, такі як ширина смуги чи максимальна швидкість.

Таблиця 2.1 ілюструє можливі варіанти стану трафіку, де:

- MTLCR визначає рівень зайнятості смуги транспортними засобами;
- TLIR показує фактичну інтенсивність трафіку в порівнянні з максимально можливою.

Таблиця 2.1 – Визначення стану трафіку на ділянці дороги за двома показниками: завантаженості (MTLCR) та інтенсивності руху (TLIR)

Значення показників	Стан
$MTLCR \to 0$ $TLIR \to 0$	Відсутні транспортні засоби на дорозі (дорога не завантажена і транспортні засоби не рухаються)
$MTLCR \to 1$ $TLIR \to 0$	Затор (дорога максимально завантажена і транспортні засоби не рухаються)
$MTLCR \to 1$ $TLIR \to 1$	Дорога максимально завантажена і транспортні засоби рухаються з максимальною дозволеною швидкістю

Як видно з таблиці, не може існувати варіант, коли MTLCR наближається до нуля, а TLIR — до одиниці. Така ситуація є фізично неможливою, оскільки за відсутності транспортних засобів (MTLCR → 0) інтенсивність руху (TLIR) повинна також спрямовуватися до нуля. Це підтверджується формулою (2.7), яка описує взаємозв'язок між цією двоїстою характеристикою.

Вдосконалення показника MTLCR:

На основі проведеного аналізу був удосконалений метод розрахунку показника MTLCR для більш точного визначення щільності дорожнього трафіку, який враховує вплив ширини транспортних засобів та перспективи

на значення цього показника. Це дозволяє зробити оцінку більш точною і надійною.

Введення показника TLIR:

TLIR — це новий показник, що дає можливість оцінити фактичну інтенсивність руху в порівнянні з максимально можливою інтенсивністю на ділянці дороги. Він обчислюється за допомогою актуальних значень MTLCR і середньої швидкості транспортних засобів. Таким чином, TLIR дозволяє динамічно оцінювати інтенсивність трафіку в реальному часі.

У результаті розробки системи показників MTLCR та TLIR було отримано ефективний інструмент для всебічного визначення поточного стану трафіку на досліджуваній ділянці дороги. Ця система дозволяє не тільки оцінювати рівень завантаженості і інтенсивності руху, але й дає можливість оперативно реагувати на зміни в трафіку, що є надзвичайно важливим для системи управління дорожнім рухом.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Моделювання програмного засобу для обчислення показників MTLCR та TLIR

Основним завданням програмного засобу є одночасне обчислення двох показників — MTLCR (Median Traffic Lane Coverage Ratio) та TLIR (Traffic Lane Intensity Ratio) для ділянки дороги, що повинні обчислюватися в реальному часі. Це означає, що швидкість обробки відеопотоку повинна бути не меншою за швидкість надходження відео. Одним з найбільшресурсоємних етапів цього процесу є обробка кадрів відео нейромережею.

Щоб підвищити ефективність і дозволити обчислювати систему показників для кількох ділянок дороги одночасно, після обробки відео нейронною мережею будуть виконуватися обчислення для кожної конкретної ділянки. Також необхідно створити можливість конфігурації та управління процесами обробки, а також отримання результатів, для чого доцільно розробити сервіс, що надаватиме користувачу інтерфейс у вигляді REST API.

Задачу можна поділити на чотири основні модулі, оскільки для обчислення TLIR необхідно мати значення MTLCR і середню швидкість потоку, що вимірюється окремо:

1. Модуль обчислення показника MTLCR — обчислює показник для кожної смуги руху з заданим інтервалом і зберігає актуальні дані в базіданих.

2. Модуль відслідковування автомобілів та обчислення середньої швидкості потоку — виконує безперервну обробку кожного кадру відеопотоку для обчислення актуальної швидкості руху автомобілів на кожній смузі.

3. Модуль обчислення показника TLIR — використовує дані з попередніх модулів для обчислення показника TLIR на основі середньої швидкості та значення MTLCR.

4. Модуль конфігурації та управління процесами обробки — надає користувачеві можливість конфігурувати параметри системи та отримувати результати через HTTP-запити.

Опис модулів:

1. Модуль обчислення показника MTLCR:

- Реалізує метод, описаний в попередньому розділі, для обчислення показника MTLCR для кожної смуги руху з заданим інтервалом.

- Результати обчислень зберігаються в базі даних разом з часовою відміткою.

- Для обчислень використовується натренована модель нейромережі U-Net.

2. Модуль відслідковування автомобілів та обчислення середньої швидкості потоку:

- Використовує модель YOLOv8 для відслідковування автомобілів в кожному кадрі відеопотоку.

- Для кожної смуги руху обчислюється середня швидкість автомобілів, і ці дані також зберігаються з часовими відмітками.

- Модуль працює безперервно для кожного кадру відео, що дозволяє отримувати актуальні дані про швидкість у реальному часі.

3. Модуль обчислення показника TLIR:

- Цей модуль обчислює TLIR на основі даних MTLCR та середньої швидкості, отриманих з двох попередніх модулів.

- Для кожного розрахованого показника MTLCR проводиться розрахунок TLIR, що дає повну систему показників для кожної смуги руху.

4. Модуль конфігурації та управління процесами обробки:

- Забезпечує інтерфейс для користувача для налаштування параметрів обробки відео та отримання результатів.

- За допомогою HTTP-запитів можна здійснювати управління процесами обробки та отримувати оновлені дані по стану трафіку.

Процес обчислення показників можна зобразити таким чином (Рис. 3.1):

- Дані з відеопотоку обробляються одночасно в двох напрямках: модулем обчислення MTLCR та модулем обчислення швидкості транспортних засобів.

- Для кожного кадру відео результати обробки зберігаються в базі даних з відповідними часовими відмітками.

- Модуль обчислення TLIR отримує ці дані і на основі них обчислює показник TLIR.

- Користувач може отримувати результати та налаштовувати процеси обробки через REST API.

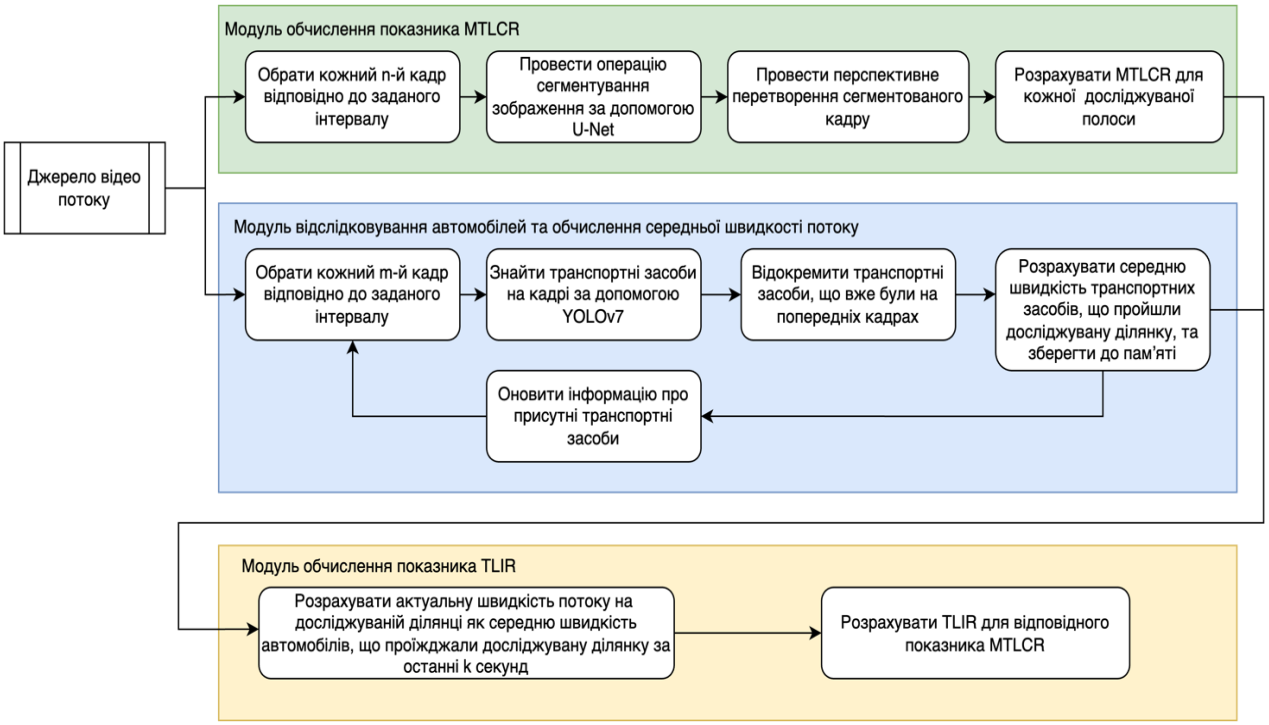

Рисунок 3.1 – Схема обчислення показників MTLCR та TLIR

Цей процес дозволяє в реальному часі отримувати точні показники для оцінки стану трафіку на ділянках дороги, що значно покращує ефективність управління дорожнім рухом.

3.2 Інструменти для розробки програмного засобу

Для реалізації програмного засобу було обрано мову програмування Python, що обумовлено кількома об’єктивними факторами. По-перше, Python є однією з найпопулярніших мов програмування в сфері штучного інтелекту та аналізу даних. Розроблено безліч бібліотек і фреймворків для роботи з штучним інтелектом, що робить цю мову ідеальним вибором для задач, пов'язаних з цією сферою. По-друге, простий і зрозумілий синтаксис Python значно полегшує розробку, зменшуючи час на написання коду і сприяючи гнучкості розробленого рішення, що особливо важливо при створенні прототипів. Для розробки програмного інтерфейсу було обрано бібліотеку aiohttp, що є асинхронним HTTP сервером для Python.

Частиною розроблених методів є обробка кадрів відеопотоку нейронними мережами: U-Net та YOLOv8. Для роботи з нейронними мережами існує два популярних фреймворки: TensorFlow та PyTorch.Оскільки розроблюване рішення не є монолітним, а складається з окремих модулів, що виконуються незалежно один від одного, використання різних технологій для кожного модуля є доцільним. Для імплементації та тренування нейронної мережі U-Net, що використовується для розрахунку MTLCR, зручніше використовувати фреймворк TensorFlow. Водночас для YOLOv8 краще підходить PyTorch, оскільки для нього вже існують натреновані моделі, які задовольняють потреби методу.

Щодо вибору бази даних, програмний засіб потребує зберігання двох типів даних: конфігурацій та результатів. Під конфігураціями маються на увазі метадані відео потоків, параметри спостережуваних ділянок дороги та інформація про запущені процеси з їх параметрами. Результатами є обчислені показники та середні швидкості транспортних засобів.

Щодо параметрів конфігурації, хоча на перший погляд здається, що доцільно використовувати SQL базу даних через потенційно фіксовану структуру конфігурацій, у цьому випадку конфігурації будуть містити різні

структури для різних типів процесів та джерел даних. Тобто, дані конфігурацій є не зовсім структурованими. Що стосується збереження результатів, вибір припадає на NoSQL базу даних з кількох причин:

- структура результатів змінюватиметься з часом, оскільки з'являтимуться нові параметри або метадані для подальшого аналізу;

- обсяг даних, що зберігаються, швидко зростатиме з додаванням нових процесів, що вимагатиме масштабування сховища, що легше здійснити в більшості NoSQL баз даних;

- дані використовуватимуться здебільшого для аналітичних запитів.

Таким чином, для зберігання даних було обрано MongoDB, що є однією з найбільш популярних NoSQL баз даних і надає всі необхідні функції для зберігання і масштабування даних в даному контексті.

3.3 Реалізація програмного засобу

Реалізація програмного засобу охоплює кілька основних етапів, пов’язаних із обробкою відеопотоку, виявленням та відслідковуванням транспортних засобів, а також обчисленням відповідних показників для аналізу дорожнього руху. Основною метою є створення ефективного інструменту для аналізу стану дорожнього руху на основі відеоданих, що дозволяє обчислювати показники MTLCR (Mean Traffic Lane Congestion Rate) та TLIR (Traffic Level Impact Rate).

Для розробки програмного засобу вибрана мова програмування Python, оскільки вона є однією з найбільш популярних у сфері штучного інтелекту та аналізу даних. Python має безліч бібліотек і фреймворків, що спрощують розробку, зокрема для роботи з нейронними мережами, що є важливою частиною цього проєкту.

Програмний засіб розділений на кілька модулів, кожен з яких виконує конкретну задачу в обробці даних. Один з основних модулів — це обчислення показника MTLCR. Цей модуль використовує нейронну мережу U-Net для сегментації зображень та виявлення автомобілів на дорозі. Для

цього кожен кадр відео, що надходить на вхід, обробляється мережею U-Net, яка виділяє об'єкти на дорозі, що дозволяє підрахувати кількість транспортних засобів на кожній смузі руху. Визначені дані використовуються для обчислення середнього показника MTLCR для кожної смуги. Результати цього процесу зберігаються в базі даних разом з часовою відміткою та іншими параметрами.

Другим важливим компонентом є модуль для відслідковування автомобілів та обчислення середньої швидкості потоку. Для цього використовується модель YOLOv8, яка є однією з найбільш точних для реального часу виявлення об'єктів на зображеннях. Кожен автомобіль, що потрапляє в кадр, відслідковується, і на основі цього обчислюється середня швидкість транспортних засобів на кожній смузі. Цей показник також зберігається в базі даних.

Третім важливим компонентом є модуль обчислення показника TLIR. Цей показник розраховується на основі даних, отриманих з попередніх двох модулів. Після того, як MTLCR та середня швидкість потоку для кожної смуги визначені, ці дані використовуються для обчислення TLIR, який відображає рівень впливу транспортного потоку на рух по дорозі.

Всі ці модулі працюють разом, формуючи єдину систему, яка здійснює обробку відеопотоку в реальному часі. Завдяки асинхронному підходу, реалізованому за допомогою бібліотеки `aiohttp`, система може ефективно обробляти відеодані та виконувати обчислення без значних затримок.

Збереження даних здійснюється в базі даних MongoDB. Вибір цієї бази даних був обумовлений її можливостями для роботи з великими об'ємами даних, що швидко зростають. Для зберігання результатів, таких як показники MTLCR, TLIR та середні швидкості, NoSQL підхід дозволяє легко масштабувати систему за потреби. MongoDB також забезпечує зручний доступ до результатів для подальшого аналізу.

Програмний інтерфейс реалізовано через асинхронний HTTP сервер, що дозволяє користувачам налаштовувати параметри відеопотоку, смуги

руху та інші конфігурації через простий інтерфейс. Це забезпечує зручність для кінцевих користувачів, дозволяючи їм запускати процеси, моніторити їх та отримувати результати без необхідності взаємодіяти безпосередньо з кодом програми.

Загалом, програмний засіб надає потужний інструмент для аналізу дорожнього руху в реальному часі, комбінуючи передові методи машинного навчання та ефективну архітектуру програмного забезпечення для обробки великих обсягів відео даних.

3.3.1 Модуль обчислення показника MTLCR

Як випливає з назви, цей модуль призначений для обчислення показника MTLCR (Mean Traffic Lane Congestion Rate), згідно з методикою, описаною в попередньому розділі. Він приймає відеофайл і конфігурацію для обробки зображення, виконує сегментацію дорожнього полотна за допомогою натренованої нейронної мережі U-Net, а також зберігає отримані результати у базу даних.

Основним компонентом програми є Python-скрипт, який приймає два аргументи: посилання на відео та конфігурацію обробки зображення у форматі JSON. Конфігурація містить інформацію про ділянки дороги, для яких потрібно обчислити показники, а також необхідний інтервал для цих обчислень. Кожна ділянка описується такими полями, як ідентифікатор, назва та координати меж. Після отримання параметрів конфігурації вони передаються класу VideoProcessor, разом з натренованою моделлю нейронної мережі для виконання сегментації.

Цей компонент реалізовано у вигляді окремого скрипту, тому необхідно забезпечити взаємодію з процесом, який викликає цей скрипт. Існує два способи виведення результатів обчислення:

1. JSON-формат до стандартного потоку виводу — цей варіант зручний, коли необхідно протестувати скрипт окремо від інших компонентів або коли система не має доступу до бази даних.

2. Запис до колекції в базі даних MongoDB — цей спосіб є більш ефективним, оскільки не потребує додаткових зусиль для обробки результатів. Вибір способу залежить від того, чи передано з бази даних параметри під час виконання скрипту.

Як було зазначено, для сегментації зображень використовується модель U-Net, яка на вхід приймає зображення розміру 256x256 пікселів і повертає сегментоване зображення того ж розміру. Кожен піксель сегментованого зображення містить значення:

- 0 — піксель належить до дороги;

- 1 — піксель не належить до дороги.

Для тренування мережі було використано дані з сервісу livetraffic.eu, який надає доступ до відеокамер із кількох країн Європи. Оскільки для обчислення MTLCR використовується сегментація дороги (на відміну від TLCR, де сегментуються автомобілі), створити датасет для тренування автоматично за допомогою фіксації руху на відео не можна. Тому датасет для тренування створювався вручну за допомогою інструменту Image Labeling Toolbox на платформі Supervisely.

Після натренування моделі, параметри моделі зберігаються у файл, який використовується для імпорту в клас VideoProcessor.

Клас VideoProcessor має метод process_video, який викликається після ініціалізації об'єкта цього класу. Процес обчислення показників для кожної ділянки дороги виглядає так:

1. Вибирається крайній кадр у відеопотоці.

2. Виконується сегментація цього зображення.

3. Для кожної досліджуваної ділянки дороги виконується перспективне перетворення.

4. Для кожної ділянки дороги обчислюється показник MTLCR, результат якого записується в базу даних і виводиться в потік.

В таблиці 3.1 наведено опис методів класу VideoProcessor та глобальних функцій, які використовуються в розробленому програмному компоненті.

Таблиця 3.1 – Опис методів класу VideoProcessor та глобальних функцій, що він використовує

Клас	Метод	Опис
Video Processo	process_video	Метод process_video запускає обробку відеопотоку, виконуючи наступний алгоритм: 1.Захоплення відеопотоку: Спочатку за допомогою класу VideoCapture з бібліотеки OpenCV здійснюється захоплення відеопотоку. Це дозволяє отримати доступ до окремих кадрів відео для подальшої обробки. 2. Видобуток та збереження метаданих відео: З відеофайлу виймаються метадані, такі як розміри кадру (ширина та висота в пікселях) і частота кадрів (кількість кадрів на секунду). Ці дані будуть використані для коректної обробки і перерахунку інтервалів. 3. Переведення інтервалу з секунд у кількість кадрів: Інтервал обчислення показника, переданий у конфігурації, переводиться з секунд у кількість кадрів, з урахуванням частоти кадрів, отриманої з метаданих відео. Це дозволяє точно визначити, які кадри повинні бути оброблені.

Таблиця 3.1 (Продовження)

	process_frame	Метод process_frame обробляє переданий кадр за допомогою наступного алгоритму: 1. Зміна розміру кадру: Першим кроком є зміна розміру кадру до 256x256 пікселів. Це необхідно для того, щоб підготувати зображення до обробки нейронною мережею, яка працює з вхідними зображеннями цього стандартного розміру. 2. Сегментація зображення: Після зміни розміру кадру виконується сегментація зображення за допомогою нейронної мережі U-Net. Мережа обробляє зображення і визначає, які пікселі належать до дороги, а які — до інших елементів (наприклад, тротуари або інші перешкоди). Результатом є сегментоване зображення, де кожен піксель має значення, що вказує на належність до дороги чи інших об'єктів. 3. Обчислення MTLCR для кожної ділянки: Після того як сегментація зображення завершена, для кожної ділянки дороги, переданої у конфігурації (інформація про межі ділянки), викликається функція calc_mtlcr. Ця функція обчислює показник MTLCR для кожної ділянки дороги, використовуючи сегментоване зображення. Результати обчислення для кожної ділянки потім зберігаються в базі даних або виводяться через стандартний потік виводу, залежно від параметрів конфігурації.

Таблиця 3.1 (Продовження)

Глобальний контекст	calc_mtlcr	Метод calc_mtlcr обчислює MTLCR для переданого кадру за допомогою наступного алгоритму: 1. Нормалізація координат меж досліджуваних ділянок: Спочатку необхідно нормалізувати координати меж ділянок дороги згідно з розмірами сегментованого зображення. Оскільки межі доріг, передані в конфігурації, визначаються у координатах оригінального відео, вони повинні бути перераховані відповідно до розміру сегментованого кадру (256x256 пікселів). 2. Перспективне перетворення: Для кожної ділянки дороги виконується перспективне перетворення зображення за допомогою функції transform_perspective. Це дозволяє коректно відобразити досліджувану ділянку дороги, враховуючи перспективу та геометрію зображення. 3. Операція згортання: Після виконання перспективного перетворення на зображенні застосовується операція згортання, яка викликає функцію reduce_transformed_mask. Ця операція зменшує область, що не є частиною дороги, і дозволяє зберегти тільки важливу інформацію для подальшого обчислення. 4. Розрахунок MTLCR: В кінці виконується розрахунок значення MTLCR за допомогою методу calc_mtlcr_by_reduced_mask. Цей метод використовує зменшену маску, яка містить тільки потрібні дані (дорогу) після виконання попередніх кроків. Розраховане значення MTLCR потім зберігається у базі даних або передається через стандартний потік виводу, в залежності від конфігурації.
	transform_perspective	Виконує перспективне перетворення кадру за допомогою інструментів бібліотеки OpenCV
	reduce_transformed_mask	Виконує операцію зрогтання кадру у одновимірний масив відповідно до кроку, описаного у методі обчислення MTLCR
	calc_mtlcr_by_reduced_mask	Розраховує значення MTLCR за даними з одновимірного масиву

У таблиці. 3.2 наведена схема колекції. Ця схема дозволяє зберігати всі важливі дані, необхідні для подальшого аналізу та перевірки точності обчислень показника MTLCR, а також для організації і структурування великих обсягів відео-даних..

Таблиця 3.2 — Опис колекції mtlcr_results

Назва стовпця	Опис	Тип даних
process_id	ID процесу обчислення MTLCR	UUID
parent_process_id	ID процесу ініціатора	UUID
created_at	Часова відмітка розрахунку MTLCR	TIMESTAMP
mtlcr	Значення розрахованого MTLCR	DOUBLE

3.3.2 Реалізація модуля обчислення середньої швидкості потоку

Цей модуль також працює з даними відео, що знімають дорожній трафік, проте його основні функції відрізняються від попереднього. Головне завдання цього компоненту — забезпечити безперервне оновлення інформації про поточну швидкість потоку на визначених смугах руху. Щоб здійснити розрахунок середньої швидкості потоку, необхідно здійснювати постійне спостереження за автомобілями на відео, використовуючи нейронну мережу для ідентифікації і обчислення їхніх швидкостей. Для цих цілей було застосовано нейронну мережу YOLOv8, яка забезпечує реальний моніторинг об’єктів у відеопотоці, якщо для цього є необхідні обчислювальні ресурси. Важливо відзначити, що ми не тренуємо мережу самостійно, а використовуємо вже натреновану модель, здатну виявляти різні об'єкти, зокрема транспортні засоби [9].

Як і у попередньому модулі, цей компонент реалізовано як Python-скрипт, який приймає два аргументи: посилання на відео та конфігурацію обробки зображень у форматі JSON. Конфігурація містить такі самі параметри, що й у компоненті для обчислення середньої кількості

транспортних засобів на смузі (MTLCR), як-от ідентифікатор, назва та координати меж. Однак в цьому модулі додаються додаткові поля, які описують кожну ділянку: довжина та ширина, що необхідні для точного розрахунку швидкості потоку. Наступним кроком є передача конфігураційних параметрів до класу `VideoProcessor`, разом з використанням моделі нейронної мережі для здійснення відслідковування.

Основними елементами даного програмного компоненту є три ключові класи:

1. LaneLocator, що відповідає за коректне співвідношення транспортних засобів з визначеними смугами руху на кадрах;
2. ObjectTracker, який зберігає інформацію про об’єкти, що відслідковуються;
3. SpeedTracker, що здійснює обчислення швидкості транспортних засобів.

Після ініціалізації відеопотоку система починає постійну обробку кадрів за допомогою нейронної мережі, що дозволяє отримувати дані про швидкості автомобілів в реальному часі. Відмінність цього модуля від попереднього полягає в тому, що в ньому неможливо вибірково обробляти кадри відео з певним інтервалом, визначеним у конфігураційному файлі. Це пов’язано з тим, що метод відслідковування заснований на виявленні об’єктіводного класу на послідовних кадрах, і точність ідентифікації залежить від частоти обробки кадрів. Також, як і в попередньому модулі, результати можуть бути збережені в базі даних для подальшого аналізу, а також виведені в потік у форматі JSON.

У таблиці 3.3 представлена структура основних класів та глобальних функцій, які використовуються в модулі для обчислення швидкості транспортних засобів.

Таблиця 3.3 – структура класів та глобальних функцій модуля обчислення швидкості транспортних засобів

Клас	Метод	Опис
SpeedTracker	process_video	Ця функція відповідає за ініціацію обробки відеопотоку та виконує наступні етапи: 1. Використовує клас VideoCapture з бібліотеки OpenCV для захоплення відеопотоку. 2. Збирає метадані потоку, такі як розміри кадрів у пікселях та частоту їх відтворення. 3. Для кожного окремого кадру відеопотоку викликає метод process_frame для подальшої обробки.
1)	process_frame	Ця функція здійснює обробку кожного кадру відеопотоку, дотримуючись наступного алгоритму: 1. Якщо оригінальний розмір кадру перевищує 720x1080 пікселів, зменшує його до зазначених розмірів. 2. Використовує нейронну мережу для обробки кадру та отримує координати меж знайдених об'єктів. 3. Застосовує метод get_bounding_boxes, щоб відфільтрувати об'єкти, що належать до цікавих класів (транспортні засоби). 4. Оновлює об'єкт ObjectTracker, додаючи нові знайдені транспортні засоби та отримуючи ті, що були виявлені на попередніх кадрах. 5. Для кожного транспортного засобу, який був присутній на попередніх кадрах, виконує метод process_speed, щоб обчислити його швидкість. 6. Якщо швидкість вдалося обчислити, записує результат до бази даних для подальшого використання.
	get_bounding_boxes	Відфільтровує об’єкти, що відповідають класам транспортних засобів

Таблиця 3.3 (Продовження)

	process_speed	Ця функція обчислює швидкість транспортних засобів за допомогою наступного алгоритму: 1. Використовує метод get_lane з сервісу LaneLocator, щоб перевірити, чи знаходиться об'єкт у межах однієї з досліджуваних ділянок. 2. За допомогою інформації про частоту кадрів відеопотоку та номер обробленого кадру визначає фактичний час, коли був оброблений поточний кадр. 3. Перевіряє, чи перетинав транспортний засіб протилежну сторону досліджуваної ділянки. 4. Якщо транспортний засіб дійсно перетинав протилежну сторону, обчислює його швидкість, використовуючи час перетину цієї сторони. 5. Якщо перетин не відбувся, зберігає інформацію про перетин сторони досліджуваної ділянки для подальшої обробки.
ObjectTracker	update	Оновлює дані про об’єкти на відео та повертає ті об’єкти, які були зафіксовані на попередніх кадрах.
LaneLocator	get_lane	Перевіряє, чи належать передані координати присутнім ділянкам дороги, та повертає дані про ділянку, якщо така була знайдена.

У Таблиці 3.4 представлена схема колекції, яка використовується для збереження даних про обчислені значення показників швидкості транспортних засобів.

Таблиця 3.4 — Опис колекції speed_results

Назва стовпця	Опис	Тип даних
process_id	ID процесу обчислення MTLCR	UUID
parent_process_id	ID процесу ініціатора	UUID
created_at	Часова відмітка розрахунку MTLCR	TIMESTAMP
speed	Обчислене значення швидкості	INT

3.3.3 Реалізація модуля обчислення показника TLIR

Модуль обчислення показника TLIR є важливою частиною автоматизованої системи моніторингу, хоча за обсягом він є одним із найменших серед розроблених компонентів. Однак, його роль у системі надзвичайно важлива, оскільки він відповідає за обчислення показника TLIR, який є ключовим для оцінки ефективності дорожнього потоку, зокрема в контексті безпеки дорожнього руху та управління трафіком.

Цей модуль призначений для роботи з даними про швидкість руху транспорту, отриманими з попереднього етапу, який займається обчисленням швидкості транспортних засобів. За допомогою актуальних даних про швидкість транспортних засобів на досліджуваній ділянці дороги, модуль здійснює обчислення показника TLIR для кожного отриманого MTLCR, що дозволяє здійснювати комплексну оцінку стану дорожнього руху в реальному часі.

Процес роботи модуля починається з того, що Python скрипт, який реалізує цей компонент, виконує пошук актуальних значень для MTLCR, швидкостей транспорту і максимально дозволеної швидкості на ділянці дороги. Ці дані отримуються з бази даних, в якій вони зберігаються в результаті роботи попередніх модулів. Для пошуку даних використовуються параметри, що визначають інтервал часу, в межах якого шукаються актуальні значення.

Після того як всі необхідні дані будуть отримані, модуль обчислює значення TLIR на основі розробленого методу, що враховує поточну

швидкість транспорту, максимально дозволену швидкість та інші параметри, пов'язані з оцінкою ефективності руху на конкретній ділянці дороги. Метод обчислення TLIR, який був розроблений в рамках цієї роботи, дозволяє отримувати точні та актуальні показники для подальшої обробки та аналізу.

По завершенню обчислення показника TLIR, отримане значення разом з відповідним значенням MTLCR зберігаються в колекції composed_results для подальшої обробки та аналізу. Ця колекція є важливою частиною бази даних, де зберігаються всі результати, отримані в процесі роботи системи. Її структура представлена у Таблиці 3.5, що дозволяє організувати дані для подальшої роботи з ними та забезпечує їх зручне використання дляприйняття управлінських рішень.

Таким чином, цей модуль виконує важливу роль у забезпеченні своєчасного моніторингу та оцінки стану дорожнього руху за допомогою використання показників TLIR і MTLCR, що дозволяє оперативно реагувати на зміни ситуації на дорогах і забезпечувати більш ефективне управління дорожнім рухом.

Таблиця 3.5 — Опис колекції composed_results

Назва стовпця	Опис	Тип даних
process_id	ID процесу обчислення системи	UUID
video_id	ID відеопотоку	UUID
lane_id	ID досліджуваної ділянки	UUID
created_at	Часова відмітка розрахунку MTLCR	TIMESTAMP
mtlcr	Значення розрахованого MTLCR	DOUBLE
tlir	Значення розрахованого TLIR	DOUBLE

3.3.4 Розробка модуля конфігурації та моніторингу

Модуль конфігурації та моніторингу є важливою частиною автоматизованої системи моніторингу відеопотоку. На відміну від попередніх двох модулів, які займалися обробкою відео та обчисленням показників, цей компонент реалізує HTTP сервіс на мові Python з використанням асинхронної бібліотеки aiohttp. Цей сервіс надає API, за допомогою якого можна конфігурувати процеси обробки відео для обчислення показників MTLCR та TLIR, керувати цими процесами в реальному часі, а також отримувати результати обчислень.

Основна мета цього модуля – забезпечити зручний інтерфейс для взаємодії з системою, налаштування та контролю роботи компонентів, а також отримання результатів обробки відео, зокрема обчислення показників дорожнього руху. Залежно від задачі, цей модуль дозволяє користувачеві налаштувати параметри, що стосуються джерела відео та ділянок для аналізу, а також моніторити та управляти процесами обробки відео.

Усі обробники запитів, реалізовані в цьому сервісі, можна поділити на три основні групи:

1. Конфігурація джерел відео: Запити цієї групи дозволяють налаштовувати джерела відео, що використовуються для аналізу дорожнього руху. Вони включають додавання нових джерел відео, а також налаштування параметрів обробки відеопотоку для подальшого використання при обчисленні показників MTLCR та TLIR. У цьому випадку також є можливість конфігурації ділянок на відео, що підлягають дослідженню (наприклад, конкретні смуги руху на дорожньому перехресті або іншій ділянці дороги).

2. Управління процесами обробки відео: Запити цієї групи надають можливість запускати, зупиняти та моніторити процеси обробки відео. Вони дозволяють управляти процесами, що відповідають за обчислення показників швидкості (MTLCR) та TLIR, а також отримувати інформацію про статус поточних процесів.

3. Виймання результатів обробки відео: Запити цієї групи відповідають за отримання результатів обробки відео. Користувач може отримати дані про обчислені показники MTLCR та TLIR, що були отримані в результаті аналізу потоку. Це дозволяє оперативно реагувати на ситуацію на дорозі, здійснювати моніторинг та аналіз стану дорожнього руху в реальному часі.

Перелік запитів, які надає кожна з груп, представлений у Таблиці 3.6. У цій таблиці наводиться детальна інформація про доступні API-методи, їх параметри та функціональні можливості, що дозволяють налаштовувати і управляти процесами обробки відео, а також отримувати результати.

Цей модуль є важливим інструментом для ефективного керування системою, оскільки дозволяє користувачеві налаштовувати роботу системи безпосередньо через HTTP API. Він забезпечує гнучкість в конфігурації, що необхідна для ефективної обробки відео та обчислення важливих показників, що використовуються для моніторингу дорожнього руху.

Таблиця 3.6 — Запити що відносяться до конфігурації відео

Метод	Path	Опис призначення
GET	/videos	Отримати присутні у базі даних конфігурації відео
POST	/videos	Додати конфігурацію відео до бази даних
GET	/videos/:video_id	Додати конфігурацію відео до бази даних
DELETE	/videos/:video_id	Отримати конфігурацію відео з бази даних за ID
POST	/videos/:video_id/lanes	Видалити конфігурацію відео з бази даних за ID
DELETE	/videos/:video_id/lanes/:lane_id	Додати досліджувану ділянку до відео присутнього у базі даних

Запити другої групи відповідають за управління процесами обчислення показників MTLCR та TLIR. Ці запити дозволяють запускати та зупиняти обробку відеопотоку, моніторити стан процесів і отримувати інформацію про статус поточних розрахунків.

Таблиця 3.7 – Запити що відносяться до управління процесами обчислення показників MTLCR та TLIR

Метод	Path	Опис призначення
POST	/videos/{video_id}/process	Почати процес обчислення показників для сконфігурованого відео за ID
GET	/processes	Отримати дані про присутні процеси обчислення показників
POST	/processes/{process_id}/stop	Зупинити процес обчислення показників за ID

Розглянемо детальніше процес запуску обчислення показників у системі. Ключовим елементом є запит, що ініціює обробку даних, який обробляється за допомогою методу `start_processing`. У цій реалізаціїнайбільш ресурсоємні обчислення були винесені в окремі скрипти, що дозволяє їх виконання в паралельних процесах. Такий підхід значно підвищує ефективність системи, особливо при роботі з великими обсягами даних і складними алгоритмами. У випадку побудови масштабованої автоматизованої системи найкращим рішенням було б впровадження спеціалізованого сервісу, який не лише відслідковує процес виконання завдань, а й забезпечує гнучке управління ними.

У поточній реалізації процеси запускаються як підпроцеси через функцію `create_subprocess_exec` із бібліотеки `asyncio`, що також забезпечує керування ними. Таким чином, керування всіма процесами реалізовано інструментами цієї ж бібліотеки. При надходженні запиту на обробку, система конфігурації та моніторингу виконує певний ланцюг дій.

Спершу генерується ідентифікатор основного процесу та дочірніх процесів, зокрема для обчислення таких показників, як MTLCR та швидкість транспортних засобів. Ці ідентифікатори дозволяють відстежувати належність дочірніх процесів до єдиного комплексу обробки відеоінформації. Після цього виконується послідовний запуск окремих скриптів для

обчислення кожного з необхідних параметрів, включаючи показники швидкості та TLIR.

Інформація про всі працюючі процеси зберігається в базі даних, що дозволяє системі відстежувати їх поточний статус. У разі завершення будь-якого з процесів обчислення (наприклад, MTLCR або швидкості), система припиняє обчислення TLIR, оскільки неповнота або некоректність даних може призвести до недостовірних результатів. Це важливо для забезпечення консистентності й точності обчислюваних показників.

При запиті на зупинку процесу система завершує роботу дочірніх процесів у тій послідовності, в якій вони були запущені, після чого оновлює відповідні записи у базі даних.

На Рис.3.2 зображено схему взаємодії між сервісом конфігурації, моніторингу та підпроцесами, що беруть участь у обробці відео. Пунктирна лінія, що з'єднує процеси обчислення MTLCR і швидкості, вказує на можливість передавання результатів через потоки виводу. Це дозволяє уникнути зайвого звернення до бази даних, яке є необхідним для процесу обчислення TLIR, оскільки для цього потрібні дані про швидкість транспортних засобів та значення MTLCR, які зберігаються в базі.

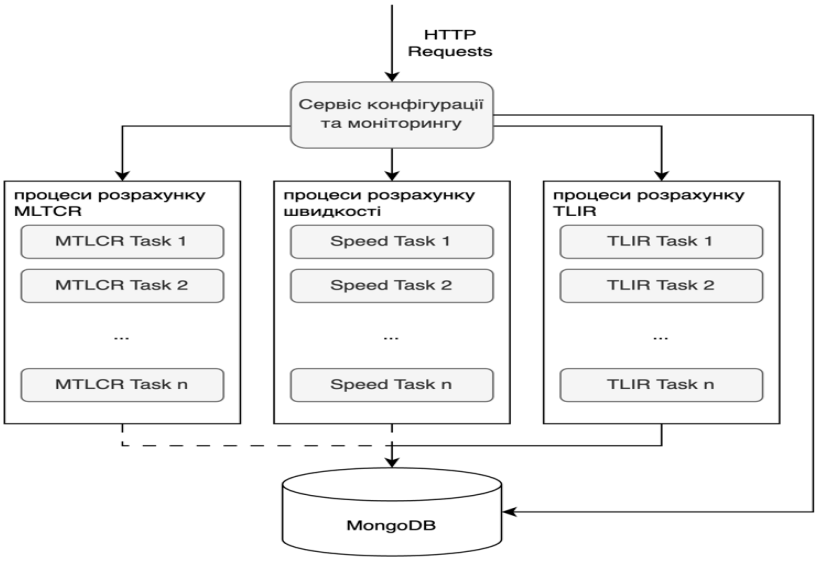

Рисунок 3.2 – Схема комунікації між модулями системи моніторингу відеопотоку

Звертаючись до налаштувань відео та досліджуваних ділянок, важливо зазначити, що колекція, яка містить параметри для конфігурації відеоресурсів і досліджуваних ділянок, описана в таблиці 3.8. Ця колекція відіграє ключову роль у збереженні й управлінні всіма параметрами, що стосуються відеопотоків і ділянок спостереження, на яких здійснюється моніторинг дорожнього руху.

Центральним елементом управління в розробленій автоматизованій системі моніторингу є веб-сервіс, який відповідає за конфігурацію та моніторинг процесів. Цей веб-сервіс є основною точкою входу в систему. Для запуску всієї системи достатньо виконати файл `main.py`, який ініціює роботу веб-сервісу на локальній машині. Завдяки простоті запуску, цей файл служить базовим інструментом для активації функціоналу системи без необхідності складних налаштувань або додаткових конфігурацій.

Крім того, система дозволяє налаштовувати логування параметрів трафіку за допомогою спеціального ключа `--log`. Цей ключ надає можливість вказати, які саме обчислені показники мають бути записані в лог.Користувач може вибрати один або декілька з наступних показників: `mtlcr`, швидкість (`speed`) або `tlir`. Це забезпечує гнучкість у зборі та збереженні необхідних даних, що є важливим для подальшого аналізу.

Для зручності відлагодження передбачено спеціальний ключ `--debug`, який активує візуалізацію процесу обчислення швидкостей транспортних засобів за допомогою нейронної мережі YOLOv8, а також результати сегментації зображень і розрахунку MTLCR за допомогою нейронної мережі U-Net. Цей режим значно полегшує аналіз роботи системи в режимі реального часу, дозволяючи візуально оцінювати, наскільки точно нейронні мережі обробляють відеопотік та визначають параметри дорожнього руху.

Приклади таких візуалізацій наведені на Рис.3.3 та Рис.3.4, де детально показано, як система визначає швидкість руху транспортних засобів та сегментацію об’єктів на відео. Ці результати демонструють точність та

надійність роботи автоматизованої системи моніторингу, що дозволяє ефективно контролювати трафік на дорожніх перехрестях.

Таблиця 3.8 — Опис колекції videos

Назва стовпця	Опис	Тип даних
id	ID конфігурації	UUID
name	Назва конфігурації	UUID
link	Посилання на ресурс з відео	STRING
lanes	Інформація про досліджувані ділянки	Lane[]
Lane.id	ID досліджуваної ділянки	UUID
Lane.name	Назва досліджуваної ділянки	STRING
Lane.coords	Координати досліджуваної ділянки	INT[][]
Lane.length	Довжина досліджуваної ділянки	DOUBLE
Lane.width	Ширина досліджуваної ділянки	DOUBLE
Lane.max_speed	Максимально дозволена швидкість на досліджуваній ділянці	INT

Усі процеси обробки, які були запущені в рамках роботи автоматизованої системи моніторингу, зберігаються та управляються через колекцію processes. Ця колекція містить детальну інформацію про кожен процес, що відстежується системою. Зокрема, до неї включаються відомості про ініціалізацію процесів, поточний стан, час початку і завершення роботи, ідентифікатори процесів, а також результати, які було отримано під час обробки відеопотоків.

Структура даної колекції представлена у вигляді таблиці, яка надає повний огляд ключових полів, необхідних для управління процесами. Зокрема, вона дозволяє відстежувати, які саме процеси знаходяться в активному стані, які вже завершені та які можуть потребувати додаткового втручання, якщо виникли збої або нестандартні ситуації під час виконання.

Схема колекції processes детально описана в таблиці 3.9, де вказані всі поля, що використовуються для зберігання інформації про процеси обробки, а також зв'язки між різними процесами, які можуть бути частиною одного

робочого потоку. Це дозволяє системі ефективно керувати всіма процесами та забезпечувати їхню синхронізацію.

Таким чином, колекція processes є критичним елементом системи, що відповідає за обробку та збереження даних про процеси, забезпечуючи надійну роботу системи моніторингу дорожнього руху, яка базується на аналізі відеопотоків із перехресть.

Таблиця 3.9 — Опис колекції processes

Назва стовпця	Опис	Тип даних
process_id	ID процесу	UUID
parent_process_id	ID процесу ініціатора	UUID
video_source	URL відео ресурсу	STRING
type	Тип процесу	ENUM
status	Статус процесу	ENUM
created_at	Часова відмітка розрахунку MTLCR	TIMESTAMP

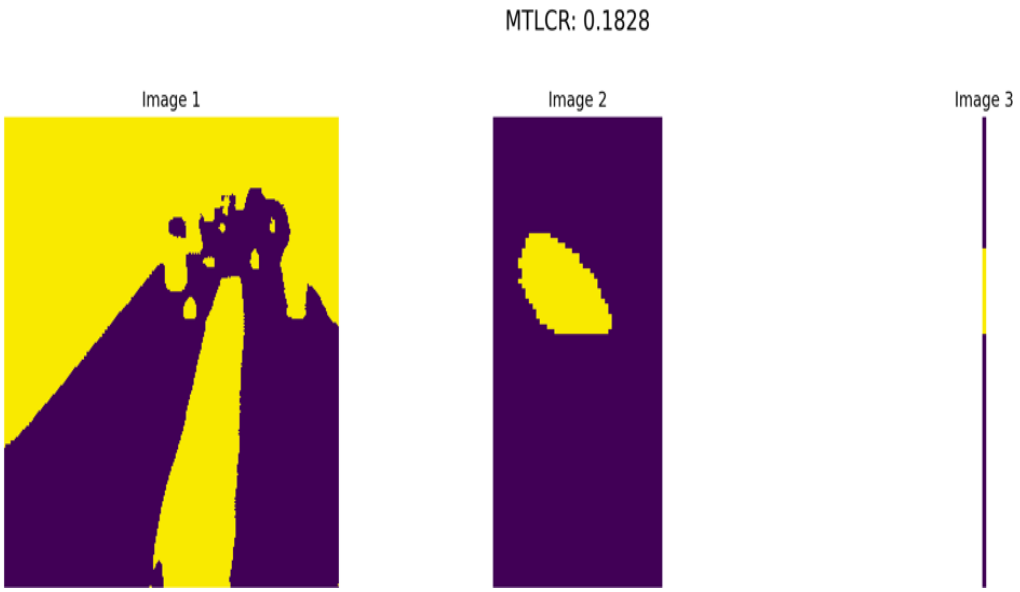

Рисунок 3.3 – Візуалізація процесу розрахунку MTLCR

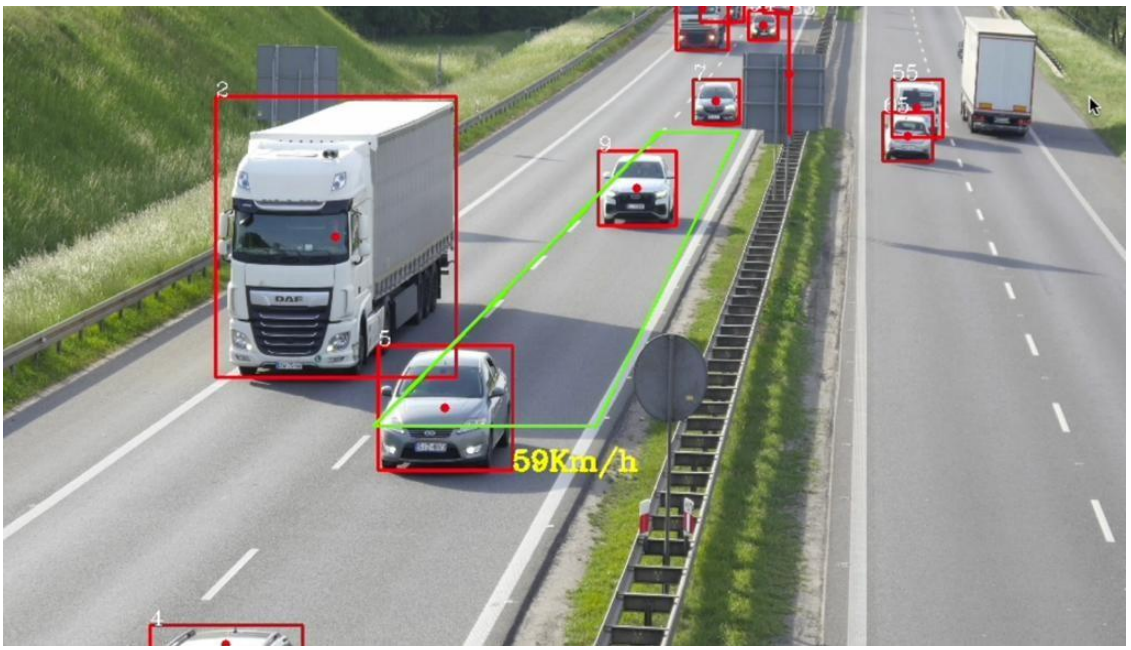

Рисунок 3.4 – Візуалізація процесу оцінки швидкості транспортних засобів

Параметри конфігурації веб-сервісу можуть бути зчитані як зі змінних середовища, так і передані безпосередньо через аргументи командної строки. Важливо, що при одночасному наданні параметрів у обох випадках, перевага буде надана значенням, переданим через аргументи командної строки. Це дозволяє зручніше налаштовувати параметри на етапі запуску програми, забезпечуючи більшу гнучкість при локальному або віддаленому розгортанні системи.

Зокрема, параметри, що стосуються підключення до бази даних, передаються через змінні середовища, що дає змогу зберігати конфіденційну інформацію, таку як логіни та паролі, окремо від інших параметрів конфігурації. Це підвищує безпеку системи, оскільки такі дані не будуть включені в командну строку або файли конфігурації, де вони можуть бути потенційно доступними для сторонніх осіб.

Приклад локального запуску веб-сервісу, що демонструє цей процес, наведено на Рис.3.5. На ньому показано, як передавати параметри через командну строку, а також як система обробляє конфігурацію, якщо значення

передані як аргументи, заміщуючи значення за замовчуванням із змінних середовища. Рис.3.5 наведено приклад локального запуску веб-сервісу.

Рисунок 3.5 – Приклад роботи програмного засобу

Розроблений програмний засіб упаковується в образ Docker, що дозволяє зручно розгортати та запускати його в різних середовищах з підтримкою технології контейнеризації Docker. Завдяки цьому, процес налаштування та розгортання програмного засобу спрощується, оскільки Docker гарантує однакове середовище для виконання незалежно від того, де запускається контейнер — чи то на локальному комп'ютері, чи на сервері.

Найбільш зручним методом для запуску контейнера є використання docker-compose. Цей інструмент дозволяє автоматизувати процес налаштування і запуску кількох контейнерів, що складають програмне середовище, забезпечуючи правильну конфігурацію всіх компонентів програми. Через docker-compose можна задати всі необхідні параметри конфігурації для запуску контейнерів, включаючи налаштування бази даних,

мережеві порти та інші залежності, що використовуються програмним засобом.

Приклад запуску контейнера за допомогою docker-compose наочно продемонстровано на Рис.3.6. На ньому наведено конфігураційний файл, в якому вказуються всі необхідні параметри для успішного розгортання програми. Такий підхід дозволяє зменшити час на налаштування середовища та забезпечити простоту у відновленні та масштабуванні програми в разі потреби. Рис.3.6 містить приклад конфігурації для запуску контейнера за допомогою docker-compose

```
version: '3'

services:
  traffic-analyzer:
    image: traffic-analyzer:latest
    environment:
      - DEBUG=false
      - LOG=tlir
      - MONGO_DB_HOST=mongodb://mongo:27017
      - MONGO_DB_NAME=database
      - MONGO_DB_USER=user
      - MONGO_DB_PASSWORD=password
    ports:
      - "8080:8080"
    volumes:
      - ./data:/app/data
~
~
~
```

Рисунок 3.6 – Файл docker-compose.yaml

У третьому розділі було представлено архітектуру програмного засобу, який призначений для обчислення показників MTLCR та TLIR на основіданих, отриманих з відеокамер у реальному часі. Згідно з запропонованою архітектурною моделлю, програмний засіб був реалізований із застосуванням мови програмування Python, що є популярним вибором для таких задач завдяки своїй простоті та широким можливостям, а також з використанням бібліотеки aiohttp для асинхронної роботи з веб-запитами.

Для обробки та аналізу відеоданих використовуються потужні бібліотеки для роботи з нейронними мережами, такі як TensorFlow та PyTorch. Ці бібліотеки забезпечують ефективне навчання та застосування глибоких нейронних мереж для таких задач, як сегментація зображень і розпізнавання об'єктів, що є необхідними для точного розрахунку показників, таких як MTLCR та TLIR.

Докладно описані основні компоненти програмного засобу та їх взаємодія між собою, що дозволяє здійснювати швидку обробку даних з відеокамер у реальному часі. Окрему увагу було приділено схемі бази даних, яка забезпечує збереження та доступ до інформації про поточні параметри трафіку та показники, що обчислюються. Окрім того, детально описано, як програмний засіб розгортається як Docker-образ, що значно полегшує його інтеграцію в різні середовища та забезпечує масштабованість і зручність у розгортанні на різних платформах.

Таким чином, у розділі розглянуто всі основні аспекти реалізації програмного засобу, від архітектури до процесу розгортання, що гарантує його ефективну роботу у різних умовах.

4 ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНОГО РІШЕННЯ

4.1 Оцінка ефективності обчислення MTLCR

В рамках дослідження порівнюємо результати обчислення значень двох показників — MTLCR і TLCR — для однакових відеокадрів. Для цього експерименту буде реалізовано як оригінальну версію методу розрахунку показника TLCR, так і його частково змінений варіант. Головною відмінністю між цими методами є процес сегментації зображення. Як було зазначено раніше, в модифікованому методі фокус ставиться на сегментацію дорожнього полотна, тоді як в оригінальному методі — на сегментацію транспортних засобів. Оскільки якість отриманих результатів може суттєво залежати від набору даних, на яких тренувалася нейронна мережа, є доцільним спробувати відокремити вплив обробки зображення нейронною мережею від обробки вже просегментованого зображення. Таким чином, поряд з оригінальним і модифікованим методами, ми вводимо проміжний підхід, при якому сегментується саме дорожнє полотно, а не транспортні засоби.

Оскільки на даний момент немає доступу до трансляцій дорожнього руху в реальному часі, ми використовуємо запис з відеокамери, розташованої на дорозі M6 у Великобританії, що був наданий для цього дослідження [32] (Рис.4.1).

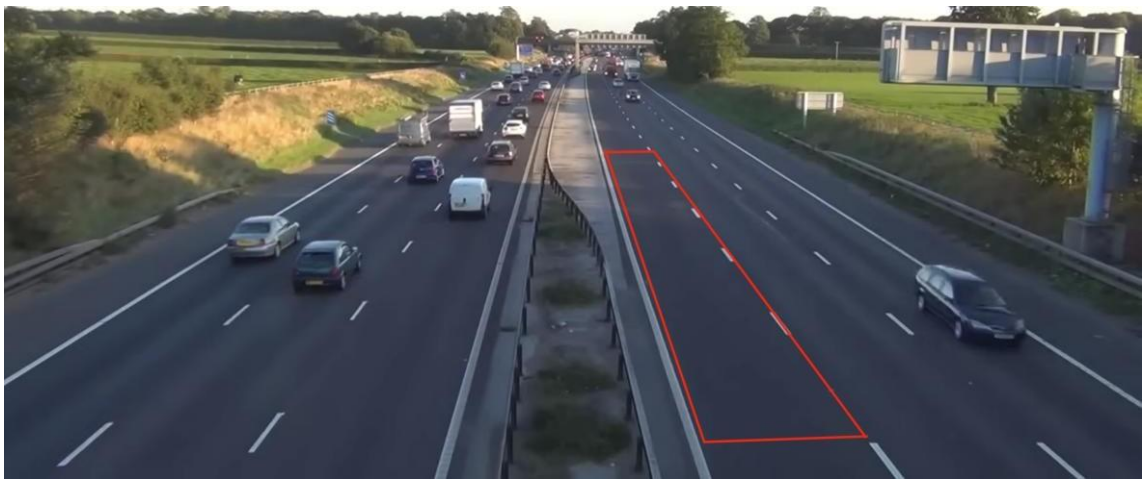

Рисунок 4.1 – Кадр з відео з досліджуваною ділянкою дороги

Протягом 10 хвилин будемо здійснювати вимірювання значень показників кожні 10 секунд за трьома описаними методами. Результати, отримані для одного з відеоканалів, представлені на Рис.4.2.

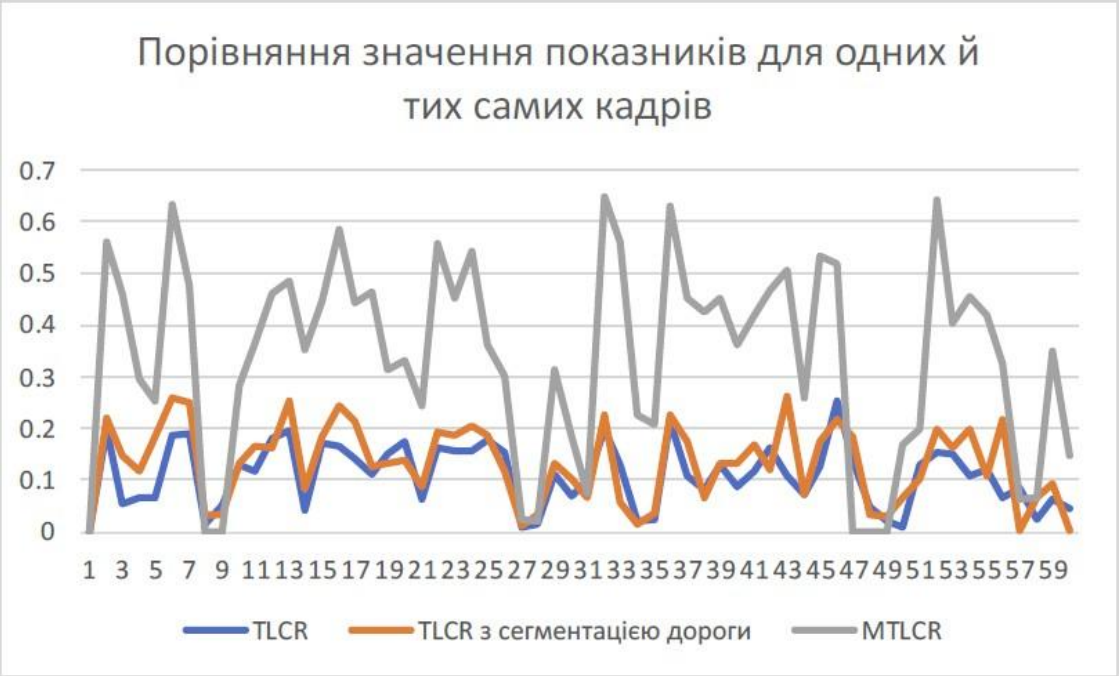

Рисунок 4.2 – Порівняння обчислених значень показників для однакових кадрів

Як видно з отриманих даних, значення MTLCR використовують значно більший інтервал значень порівняно з TLCR. Зокрема, при розрахунку MTLCR максимальні значення досягали 0,65, тоді як при обчисленні TLCR цізначення не перевищували 0,2. Крім того, можна помітити, що при використанні сегментації дороги показники TLCR в середньому були вищими, ніж при сегментації транспортних засобів. Це можна пояснити тим, що оригінальний метод, орієнтуючись на сегментацію саме транспортних засобів, не завжди коректно виділяв автомобілі на кадрі. Це сталося через обмежену здатність нейронної мережі U-Net сегментувати всі типи транспортних засобів. З іншого боку, при сегментації дороги таких проблем не виникало, тому нейронна мережа могла виявити більше об'єктів, які не належать до дорожнього полотна, включаючи автомобілі. В результаті цього

обчислене значення показника виявлялося дещо більшим, ніж у випадку сегментації транспортних засобів.

4.2 Оцінка ефективності обчислення TLIR

Для тих самих відеоматеріалів було проведено також вимірювання показника TLIR. Результати, що були отримані за допомогою цієї системи показників, представлені на Рис.4.3.

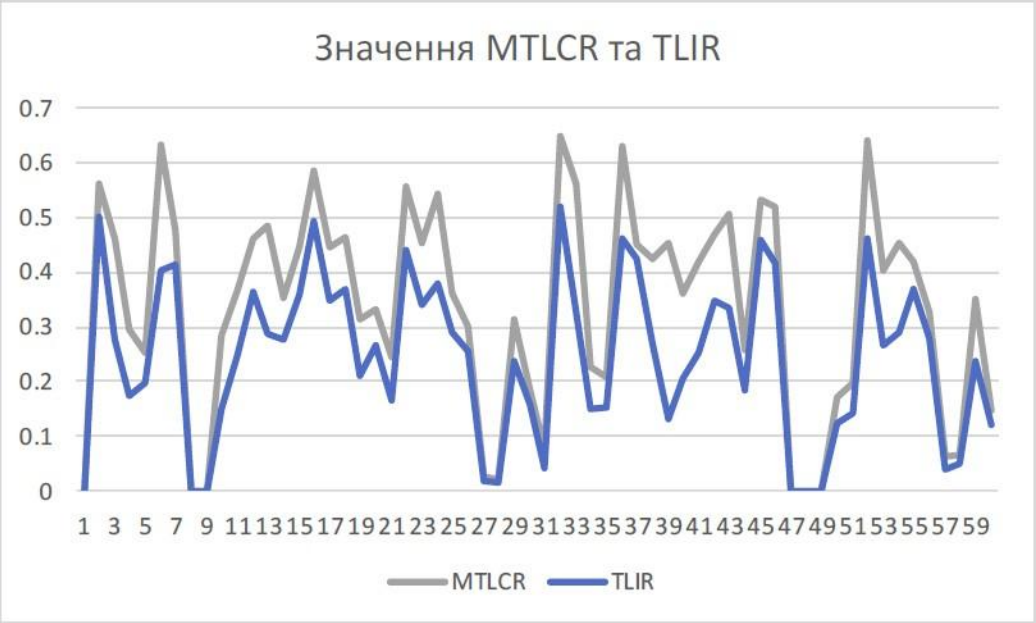

Рисунок 4.3 – Візуалізація обчисленої системи показників

Як видно з отриманих даних, значення TLIR для більшості кадрів наближаються до показників MTLCR, що свідчить про те, що транспортні засоби рухаються зі швидкістю, яка близька до максимально можливої на даній ділянці дороги. Більш детальний аналіз показує, що значення TLIR знаходяться в прямій залежності від MTLCR. Наприклад, коли значення MTLCR дорівнює нулю, TLIR також має значення нуль, що підтверджує відсутність трафіку в момент вимірювання, навіть якщо середня швидкість руху на смузі, згідно з базою даних, не є нульовою. Завдяки такій залежності, неможлива ситуація, коли значення TLIR буде більшим за значення MTLCR, оскільки цей показник прямо корелює з інтенсивністю руху.

У четвертому розділі була проведена оцінка ефективності розроблених методів та програмних засобів. Зокрема, були проведені експерименти, що включають порівняння результатів розрахунку TLCR та MTLCR для одних і тих самих кадрів. Також було проведено комплексне оцінювання системи показників MTLCR і TLIR. Результати експериментів підтвердили теоретичну основу модифікованого методу обчислення MTLCR та нового показника TLIR, що демонструє адекватність і точність запропонованого програмного засобу для вирішення задачі оцінки дорожнього трафіку на основі відеоданих.

5 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявсяза реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальноїсистеми оцінювання за 12-ма критеріями, наведеними в табл. 4.1 .

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту	Технічні та споживчі влас-тивості продукту	Технічні та споживчі властивості продукту на	Технічні та споживчі властивості продукту	Технічні та споживчі властивості продукту
5	Експлуатаційні витрати значно вищі, ніж в	Експлуатаційні витрати дещо вищі, ніж в	Експлуатаційні витрати на рівні експлуатаційни	Експлуатаційні витрати трохи нижчі, ніж в	Експлуатаційні витрати значно нижчі, ніж в
Ринкові перспективи					
6	Ринок малий і не має позитивної	Ринок малий, але має позитивну	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела	Потрібні незначні фінансові ресурси. Джерела фінансування	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуют ь ся у військово	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуют
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	5	4
2. Ринкові переваги (наявність аналогів)	4	4	5
3. Ринкові переваги (ціна продукту)	3	4	4
4. Ринкові переваги (технічні властивості)	4	4	5
5. Ринкові переваги (експлуатаційні витрати)	4	3	3
6. Ринкові перспективи (розмір ринку)	3	2	3
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	2	2	2
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	4

Сума балів	40	42	43
Середньоарифметична сума балів $СБ_c$	41,7		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [27].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» становить 41,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

5.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [27]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i \,, \tag{4.1}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *i-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k}\alpha_i = 1$;

β_i – відносне значення *i-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ні}}{I_{ai}}, \tag{4.2}$$

де $I_{ні}$ та $I_{на}$ – чисельні значення конкретного *i*-го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ні}}; \tag{4.3}$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість параметрів, що обробляються в запиті	од.	20	30	1,5	0,15
Швидкість	мс	0,5	0,1	5	0,2

обробки зображення					
Точність обробки зображення	%	92	97	1,05	0,25
Обсяг баз даних, що підлягають обробці	ГБ	128	256	2	0,25
Формування та оновлення баз даних	бал	7	8,5	1,21	0,15

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1,5 \cdot 0,15 + 5 \cdot 0,2 + 1,05 \cdot 0,25 + 2 \cdot 0,25 + 1,21 \cdot 0,15 = 2,17.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,17 рази.

5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Автоматизована система моніторингу відеопотоку дорожнього перехрестя», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами

оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=22 дні.

$$З_о = 17250{,}00 \cdot 22 / 22 = 17250{,}00 \; грн.$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	17250,00	784,09	22	17250,00
Інженер-розробник програмного забезпечення	20400,00	927,27	7	6490,91
Інженер-проектувальник автоматизованих систем	16800,00	763,64	18	13745,45
Консультант (фахівець-експерт ПДР)	18000,00	818,18	5	4090,91
Всього				41577,27

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \quad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо *Мм*=8000,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) **[**28**]**;

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 22 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 8000{,}00 \cdot 1{,}10 \cdot 1{,}15 / (22 \cdot 8) = 57{,}50$ *грн.*

$З_{р1} = 57{,}50 \cdot 5{,}00 = 287{,}50$ *грн.*

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка офісного обладнання	5,00	2	1,10	57,50	287,50
Підготовка робочого місця розробника автоматизованих систем	6,50	2	1,10	57,50	373,75
Інсталяція програмного забезпечення розробки системи	4,25	5	1,70	88,86	377,67
Формування бази даних	8,20	4	1,50	78,41	642,95
Тренування системи	5,50	3	1,35	70,57	388,13

моніторингу					
Налагодження програмних блоків	3,61	5	1,70	88,86	320,80
Тестування автоматизованої системи	16,00	1	1,00	52,27	836,36
Всього					3227,16

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_{о} + З_{р}) \cdot \frac{Н_{дод}}{100\%}, \quad (4.7)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$З_{дод} = (41577,27 + 3227,16) \cdot 10 / 100\% = 4480,44$ *грн.*

5.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_{н} = (З_{о} + З_{р} + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.8)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$Зн = (41577,27 + 3227,16 + 4480,44) \cdot 22 / 100\% = 10842,67$ *грн.*

5.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Автоматизована система моніторингу відеопотоку дорожнього перехрестя».

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^{n} H_j \cdot Ц_j \cdot К_j \; - \sum_{j=1}^{n} B_j \cdot Ц_{в\,j} \,, \qquad (4.9)$$

де H_j – норма витрат матеріалу *j*-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу *j*-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 … 1,15*);

B_j – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

$M_1 = 2 \cdot 186{,}00 \cdot 1{,}03 - 0 \cdot 0 = 383{,}16$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір VERDE Ultra	186,00	2	0	0	383,16
Папір для записів VERDE Light A5	99,00	3	0	0	305,91
Органайзер офісний VERDE OFFICE	172,00	2	0	0	354,32
Канцелярське приладдя (набір офісного працівника)	202,00	4	0	0	832,24
Картридж для принтера Canon LBP6000	1208,00	1	0	0	1244,24
Диск оптичний NewCODE CD-R	25,00	3	0	0	77,25
Flesh-пам'ять Kingston 32 GB	239,00	1	0	0	246,17
Тека для паперів CARBONIX BOX-ZX	114,00	2	0	0	234,84
Інші матеріали	165,00	1	0	0	169,95
Всього					3848,08

5.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Автоматизована система моніторингу відеопотоку дорожнього перехрестя», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \qquad (4.10)$$

де $Н_j$ – кількість комплектуючих j-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих j-го виду, грн;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 … 1,15*).

$К_в = 2 \cdot 260{,}00 \cdot 1{,}03 = 535{,}60$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Сенсори руху	2	260,00	535,60
Сенсори освітлення	2	350,00	721,00
Сенсори для вимірювання погодних умов	2	2799,00	5765,94
Відеокамера високої роздільної здатності	1	12560,00	12936,80
Всього			19959,34

5.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$В_{спец} = \sum_{i=1}^{k} Ц_i \cdot С_{пр.i} \cdot К_i \ , \qquad (4.11)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.і}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, (K_i = *1,10…1,12);*

k – кількість найменувань устаткування.

$B_{спец}$ = *26057,00 · 1 · 1,03 = 26838,71 грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Сервер ARTLINE Business R17 v14	1	26057,00	26838,71
Мережеве сховище Synology 4BAY DS423+ (DS423+)	1	21780,00	22433,40
Всього			49272,11

5.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.12)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, (K_i = *1,10…1,12);*

k – кількість найменувань програмних засобів.

$B_{прг} = 7264{,}00 \cdot 1 \cdot 1{,}02 = 7409{,}28$ *грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Середовище математичного інженерного моделювання Matchcad 12	1	7264,00	7409,28
Абонплата доступу до мережі Internet	1	350,00	357,00
Середовище VisioPRO	1	6250,00	6375,00
Всього			14141,28

5.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \qquad (4.13)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (24680{,}00 \cdot 1) / (3 \cdot 12) = 685{,}56$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер HP ProBook 4540s	24680,00	3	1	685,56

Обчислювально-графічна система програмної розробки	42888,00	3	1	1191,33
Робоче місце розробника програмного забезпечення	8599,00	5	1	143,32
Пристрій виводу текстової інформації	6500,00	4	1	135,42
Оргтехніка	8925,00	4	1	185,94
Приміщення лабораторії	520000,00	25	1	1733,33
Пристрій швидкісної передачі данних	9850,00	4	1	205,21
Всього				4280,10

5.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \qquad (4.14)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e$ = 10,98 грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,25 \cdot 160,0 \cdot 10,98 \cdot 0,95 / 0,97 = 439,20$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Обчислювально-графічна система програмної розробки	0,25	160,0	439,20
Робоче місце розробника програмного забезпечення	0,08	160,0	140,54
Пристрій виводу текстової інформації	0,25	3,0	8,24
Оргтехніка	0,32	1,3	4,57
Сервер ARTLINE Business R17 v14	0,12	160,0	210,82
Мережеве сховище Synology 4BAY DS423+ (DS423+)	0,08	150,0	131,76
Пристрій швидкісної передачі данних	0,04	150,0	65,88
Всього			1001,00

5.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов’язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з’їзди, конференції, наради, пов’язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{H_{св}}{100\%}, \quad (4.15)$$

де $H_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$B_{св} = (41577,27 + 3227,16) \cdot 20 / 100\% = 8960,89$ *грн.*

5.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_{о} + З_{р}) \cdot \frac{H_{сп}}{100\%}, \quad (4.16)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 30%.

$B_{сп} = (41577,27 + 3227,16) \cdot 30 / 100\% = 13441,33$ *грн.*

5.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{в} = (З_{о} + З_{р}) \cdot \frac{H_{ів}}{100\%}, \quad (4.17)$$

де $H_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$I_{в} = (41577,27 + 3227,16) \cdot 50 / 100\% = 22402,22$ *грн.*

5.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків;

витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_{о} + З_{р}) \cdot \frac{H_{нзв}}{100\%}, \quad (4.18)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 105%.

$B_{нзв} = (41577,27 + 3227,16) \cdot 105 / 100\% = 47044,66$ *грн.*

Витрати на проведення науково-дослідної роботи на тему «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_{о} + З_{р} + З_{дод} + З_{н} + М + К_{в} + B_{спец} + B_{прг} + A_{обл} + B_{е} + B_{св} + B_{сп} + I_{в} + B_{нзв}. \quad (4.19)$$

$B_{заг}$= *41577,27 +3227,16 +4480,44 +10842,67 +3848,08 +19959,34 +49272,11 + 14141,28 + 4280,10 +1001,00 +8960,89 +13441,33 +22402,22 +47044,66 = =244478,55 грн.*

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.20)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η =0,95.

ЗВ = 244478,55 / 0,95 = 257345,85 грн.

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження

результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів пристрою, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	100	200	250	200

N – кількість споживачів які використовували аналогічний пристрій у році до впровадження результатів нової науково-технічної розробки, приймемо 2400 осіб;

$Ц_б$ – вартість пристрою у році до впровадження результатів розробки, приймемо 45000,00 грн;

$\pm\Delta Ц_o$ – зміна вартості пристрою від впровадження результатів науково-технічної розробки, приймемо 2605,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [27]:

$$\Delta П_i = (\pm\Delta Ц_o \cdot N + Ц_o \cdot \Delta N) \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.21)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%,

а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =40%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році ϑ =*18%*;

Збільшення чистого прибутку 1-го року:

$\Delta П_1$ =(2605,00·2400,00+47605,00·100)·0,83·0,4·(1-0,18/100%)=2998043,00 грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2$ =(2605,00·2400,00+47605,00·300)·0,83·0,4·(1-0,18/100%)=5590040,04 грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3$ =(2605,00·2400,00+47605,00·550)·0,83·0,4·(1-0,18/100%)=8830036,34 грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4$ =(2605,00·2400,00+47605,00·750)·0,83·0,4·(1-0,18/100%)=11422033,38 грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (4.22)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

Т – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ =0,15;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$ПП$ =2998043,00/(1+0,15)^1+5590040,04/(1+0,15)^2+8830036,34/(1+0,15)^3+
+11422033,38/(1+0,15)^4=2606993,91+4226873,38+5805892,23+6530584,66=
=19170344,17 грн.

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \qquad (4.23)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 257345,85 грн.

$PV = k_{інв} \cdot ЗВ$= 2 · 257345,85 = 514691,69 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \qquad (4.24)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 19170344,17 грн;

PV – теперішня вартість початкових інвестицій, 514691,69 грн.

$E_{абс} = ПП - PV$ = 19170344,17 - 514691,69 = 18655652,48 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію

науково-технічної розробки:

$$E_в = \sqrt[Тж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.25)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 18655652,48 грн;

PV – теперішня вартість початкових інвестицій, 514691,69 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_в = \sqrt[Тж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+18655652,48/514691,69)^{1/4} = 1,47.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (4.26)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні d =0,11;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,35.

$\tau_{мін}$ = 0,11+0,35 = 0,46 < 1,47 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \qquad (4.27)$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 1,47 = 0,68 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалурозробки за темою «Автоматизована система моніторингу відеопотоку дорожнього перехрестя» становить 41,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,17 рази.

Також термін окупності становить 0,68 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Автоматизована система моніторингу відеопотоку дорожнього перехрестя».

ВИСНОВКИ

У ході виконання магістерської роботи було розглянуто актуальне наукове завдання пов’язані зі збільшенням ефективності (точності,швидкодії) аналізу дорожнього трафіку за даними з відеокамер у реальному часі. При цьому було отримано наступні наукові та практичні результати.

Проведено детальний аналіз існуючих рішень, починаючи від ручних методів до повністю автоматизованих методів, що використовують методи машинного навчання для виконання аналізу та дозволяють виконувати аналіз у реальному часі. На основі даних, отриманих в процесі аналізу, сформульовано задачу розробки нових та підвищення ефективності (точності, швидкодії) існуючих методів обробки даних відеокамер спостереження транспортного руху в реальному часі.

Взявши за основу показник TLCR, що визначає завантаженість ділянки дороги транспортними засобами, було розроблено його модифіковану версію MTLCR, яка підвищує точність оригінального методу та дозволяє використовувати його для аналізу трафіку на більших за розміром ділянках дороги. Завдяки зміні підходу до сегментації (сегментації дороги замість транспортних засобів) було зменшено залежність значення показника від набору даних, на якому було натреновано нейронну мережу. Завдяки перспективному перетворенню було зменшено вплив перспективи на розрахунок показника. Використання згортки двовимірного масиву у одновимірний замість розрахунку загальної кількості пікселів, що належать транспортним засобам, дозволило усунути вплив ширини транспортних засобів та ширини дороги на значення показника.

Введено новий показник TLIR (Traffic Line Intensity Ration), що визначає інтенсивність руху транспортних засобів ділянкою дороги з використанням актуального значення показника MTLCR та актуальноїсередньої швидкості транспортних засобів для досліджуваної ділянки. Для

розрахунку TLIR було розроблено алгоритм для оцінки швидкостей транспортних засобів на ділянці дороги.

Введено систему показників MTLCR та TLIR, за якою можна точно та всебічно визначити стан трафіку: завантаженість ділянки дороги та інтенсивність руху транспортних засобів нею.

Запропоновано архітектуру програмного засобу для обчислення показників MTLCR та TLIR за даними з відеркамер. Відповідно до запропотованої архітектури, розроблено програмний засіб з використанням мови програмування Python, бібліотеки aiohttp для написання web-сервісу, TensorFlow та PyTorch для роботи з нейронними мережами.

Експерименти, проведені з використанням розробленого програмного засобу підтвердили правильність теоретичного аспекту розроблених методів обчислення показників. Запропонована модифікація MTLCR ефективно працює для більших ділянок дороги, на відміну від оригінального методу та ефективніше використовує діапазон значень від 0 до 1. Також проведені експерименти підтвердили можливість програмного засобу працювати у реальному часі при оцінці швидкостей транспортних засобів, на основі яких потім обчислюється показник TLIR.

ПЕРЕЛІК ПОСИЛАНЬ

1. Савастру С.В., Стеценко І.В. (2023). Методи обробки даних відеокамер спостереження транспортного руху в реальному часі. Міжвідомчий науково- технічний журнал «Адаптивні системи автоматичного управління» 2 (43), 164- 173.https://doi.org/10.20535/1560-8956.43.2023.292269

2. Савастру С.В., Стеценко І.В. Методи обробки даних відеокамер спостереження. Інженерія програмного забезпечення і передові інформаційні технології (Soft Tech-2023): матеріали V Міжнародної науково-практичної конференції молодих вчених та студентів, 19-21 грудня 2023 року, м. Київ, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», ФІОТ. С.281-284.

3. W. Liu, Y. Okubo, J. Sun and Y. Aoyama, "A 79GHz Millimeter Wave Radar Based Car and Motorcycle Counter," 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 2020, pp. 1-6, doi: 10.1109/RadarConf2043947.2020.9266604.

4. Z. Yang, F. Shi and H. Liang, "A Portable Traffic Counting, Speed Estimation, and Classification Terminal Using IR-UWB Radar," in IEEE Sensors Journal, vol. 22, no. 13, pp. 13365-13374, 1 July1, 2022, doi: 10.1109/JSEN.2022.3181215.

5. Y. -s. Choi, P. Zaijun, S. -w. Kim, T. -h. Kim and C. -b. Park, "Salient Motion Information Detection Technique Using Weighted Subtraction Image and Motion Vector," 2006 International Conference on Hybrid Information Technology, Cheju, Korea (South), 2006, pp. 263-269, doi: 10.1109/ICHIT.2006.253497.

6. S. Zhang and F. Stentiford, "Motion Detection using a Model of Visual Attention," 2007 IEEE International Conference on Image Processing, San Antonio, TX, USA, 2007, pp. III - 513-III - 516, doi: 10.1109/ICIP.2007.4379359.

7. Black, M.; Anandan, P. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. ComputerVision and Image Understanding (1996), 63(1), pp. 75–104.

8. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on ComputerVision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788,doi: 10.1109/CVPR.2016.91.

9. Mao, Q.C., Sun, H.M., Zuo, L.Q. and Jia, R.S. (2020). "Finding every car: a traffic surveillance multi-scale vehicle object detection method". Applied Intelligence, p. 12.

10. Wang, Chien-Yao & Bochkovskiy, Alexey & Liao, Hong-yuan. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 10.48550/arXiv.2207.02696.

11. Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Електронний ресурс] // Режим доступу: https://github.com/ultralytics/ultralytics

12. Zhang, Y.; Sun, Y.; Wang, Z.; Jiang, Y. YOLOv7-RAR for UrbanVehicle Detection. Sensors 2023, 23, 1801. https://doi.org/10.3390/s23041801

13. Yong-Kul Ki and Doo-Kwon Baik, "Model for accurate speed measurement using double-loop detectors," in IEEE Transactions on Vehicular Technology, vol. 55, no. 4, pp. 1094-1101, July 2006, doi: 10.1109/TVT.2006.877462.

14. D. Lhomme-Desages, C. Grand, F. B. Amar and J. -C. Guinot, "Doppler-Based Ground Speed Sensor Fusion and Slip Control for a Wheeled Rover," in IEEE/ASME Transactions on Mechatronics, vol. 14, no. 4, pp. 484-492, Aug. 2009, doi: 10.1109/TMECH.2009.2013713.

15. Dogan, Sedat & Temiz, Mahir & Külür, Sıtkı. (2010). Real Time Speed Estimation of Moving Vehicles from Side View Images from an Uncalibrated Video Camera. Sensors (Basel, Switzerland). 10. 4805-24. 10.3390/s100504805.

16. Dahl, Mattias & Javadi, Saleh. (2019). Analytical Modeling for a Video-Based Vehicle Speed Measurement Framework. Sensors. 20. 160. 10.3390/s20010160.

17. A. P. Kulkarni and V. P. Baligar, "Real Time Vehicle Detection, Tracking and Counting Using Raspberry-Pi," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 2020, pp. 603- 607, doi: 10.1109/ICIMIA48430.2020.9074944.

18. H. Zhang, M. Liptrott, N. Bessis and J. Cheng, "Real-Time Traffic Analysis using Deep Learning Techniques and UAV based Video," 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 2019, pp. 1-5, doi: 10.1109/AVSS.2019.8909879.

19. Yu, Qingying & Luo, Yonglong & Chen, Chuanming & Zheng, Xiaoyao. (2019). Road Congestion Detection Based on Trajectory Stay-Place Clustering. ISPRS International Journal of Geo-Information. 8. 264. 10.3390/ijgi8060264.

20. A. Kanungo, A. Sharma and C. Singla, "Smart traffic lights switching and traffic density calculation using video processing," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 2014, pp. 1-6, doi: 10.1109/RAECS.2014.6799542.

21. Stetsenko, I.V., Stelmakh, O. (2020). Traffic Lane Congestion Ratio Evaluation by Video Data. Advances in Intelligent Systems and Computing 1019, 172-181. Springer, Cham. https://doi.org/10.1007/978-3-030- 25741-5_18

22. Ronneberger, O., Fischer, P. and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

23. Car and truck traffic on the highway in europe poland summer day. URL: https://www.vecteezy.com/video/7957364-car-and-truck-traffic-on-the- highway-in- europe-poland-summer-day (date of access: 02.01.2024)

24. OpenCV. URL:https://docs.opencv.org/4.x/ (date of access: 02.10.2024)

25. Python3 documentation [Електронний ресурс] // Режим доступу: https://docs.python.org/3/

26. AIOHTTP documentation [Електронний ресурс] // Режим доступу: https://docs.aiohttp.org/en/stable/

27. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

28. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

29. Всеукраїнська науково-технічна конференція ФІІТА URL: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20875

30. Кисельов В., Ковтун В., Юхимчук М. Машинні методи проблемно-орієнтованого бізнес аналізу великих даних вінницького регіону. Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах». 2024. № 2. С. 59–64.

31. Yukhymchuk M. Decentralized Coordination of Temperature Control in Multiarea Premises [Text] / M. Yukhymchuk, V. Dubovoi, V. Kovtun // Complexity. – 2022. – Vol. 22. – Article ID 2588364.

32. Биков М. М., Грищук Т. В., Ковалюк О. О., Ковтун В. В., Юхимчук М. С. Модель експлуатації кіберфізичної системи в умовах впливу негативних зовнішніх факторів. Вісник Вінницького політехнічного інституту. 2023 Вип. 6. С. 30–38.

33. Дубовой В. М. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 151 – Автоматизація та комп’ютерно-інтегровані технології – Вінниця : ВНТУ, 2022 – 38с

Додатки

Додаток А
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: « Автоматизована система моніторингу відеопотоку дорожнього перехрестя »

Тип роботи: магістерська кваліфікаційна робота
(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (зазначити))

Підрозділ кафедра комп'ютерних систем управління
(кафедра, факультет (інститут), навчальна група)

Керівник Юхимчук М.С., професор кафедри КСУ
(прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	99%
Загальна схожість	1%

Аналіз звіту подібності (відмітити потрібне)

√ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Ярослав РОЇК
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

_____

_____

_____

Особа, відповідальна за перевірку _____ Володимир ДУБОВОЙ
(підпис) (прізвище, ініціали)

Експерт _____ _____
(за потреби) (підпис) (прізвище, ініціали, посада)

ДОДАТОК Б

(обов'язковий).

Технічне завдання

ВНТУ

ЗАТВЕРДЖУЮ
Зав. кафедри КСУ
д.т.н., професор
В'ячеслав КОВТУН
" 14 " 10 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістреської кваліфікаційної роботи

«Автоматизована система моніторингу відеопотоку дорожнього перехрестя»

08.33.МКР.016.00.000 ТЗ

Студент групи 2АКІТР-23м
Ярослав РОЇК
Підпис *(Ім'я та ПРІЗВИЩЕ)*

Керівник: Д.т.н., проф. каф. КСУ
Марія ЮХИМЧУК
Підпис *(Ім'я та ПРІЗВИЩЕ)*

Вінниця 2024

1. Назва та галузь застосування

1.1. Назва – Автоматизована система моніторингу відеопотоку дорожнього перехрестя

1.2. Галузь застосування – системи безпеки дорожнього руху, автоматизація аналізу дорожньої обстановки.

2.Підстава для проведення розробки.

2.1. Тема магістреської кваліфікаційної роботи затверджена наказом по ВНТУ №310 від 17.09.2024

3.Мета та призначення розробки.

3.1. Метою магістерської кваліфікаційної роботи є підвищення безпеки дорожнього руху через автоматизацію процесу моніторингу відеопотоку дорожнього перехрестя за допомогою розробленої автоматизованої системи.

4. Джерела розробки.

4.1. Магістреська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

- Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології / Уклад. В. М. Дубовой, – Вінниця : ВНТУ, 2022 – 38с.
- Савастру С.В., Стеценко І.В. Методи обробки даних відеокамер спостереження. Інженерія програмного забезпечення і передові інформаційні технології (Soft Tech-2023): матеріали V Міжнародної науково-практичної конференції молодих вчених та студентів, 19-21 грудня 2023 року, м. Київ, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», ФІОТ. С.281-284.
- W. Liu, Y. Okubo, J. Sun and Y. Aoyama, "A 79GHz Millimeter Wave Radar Based Car and Motorcycle Counter," 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 2020, pp. 1-6, doi: 10.1109/RadarConf2043947.2020.9266604.

5.Вимоги до розробки.

5.1. Перелік головних функцій:

1. Автоматичне розпізнавання транспортних засобів та інших об'єктів на відеопотоці з перехрестя.
2. Аналіз трафіку та визначення основних характеристик дорожнього руху (швидкість, інтенсивність, аварійні ситуації)
3. Візуалізація даних для моніторингу в режимі реального часу.
4. Надання інструментів для адміністратора для управління системою, перегляду статистики та формування звітів.

5.2. Основні технічні вимоги до розробки

5.2.1. Вимоги до програмної платформи:
- •Window 10;
- •Linux Ubuntu 22.04;
- •MacOS 13;
- •iOS 15;
- •Android 13.

5.2.2. Умови експлуатації системи:
- •робота на усіх типах девайсів;
- •можливість цілодобового функціонування системи;
- •дані оновлюються і є актуальними.

6. Стадії та етапи розробки.

6.1. Поянювальна записка:

1. Аналіз методів, приницпів, підходів і засобів реалізації задачі автоматизації процесами в об’єкті управління відповідно до теми кваліфікаційної роботи.
Постановка задач дослідження « 04 » вересня 2024 р.
2. Удосконалення технології прийняття рішень при автоматизації об’єкту «1» 10 2024 р.
3.Визначення техніних характеристик системи «9» 10 2024 р.
4.Розробка програмного забезпечення системи «24» 10 2024 р.

6.2. Графічні матеріали:

1. Розробка UML-діаграми «3» 11 2024 р.
2. Розробка моделі бази данних «15» 11 2024 р.
3.Тестування програмного забезпечення «22» 11 2024 р.

7. Порядок контроллю й приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «22» 11 2024 р.

7.2. Атестація проекту здійснюється на попередньому захисті. Попередній захист магістреської кваліфікаційної роботи провести до «01» 12 2024 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістреської кваліфікаційної роботи провести до «9» 12 2024р.

ДОДАТОК В
(обов'язковий).
Лістинг програмного коду

Лістинг В.1. – Модуль run_mtlcr.py

```
import argparse import json

from lane import Lane
from video_processor import VideoProcessor import sys

DEFAULT_LANE_1 = Lane(0, 'left', [(1250, 1525), (1950, 1525), (2150, 743), (2400, 743)], 70, 7, 120)
DEFAULT_LANE_2 = Lane(1, 'right', [(2800, 1525), (3400, 1525), (2800, 743), (3050, 743)], 70, 7,
120)



if

name

== " main ":

parser = argparse.ArgumentParser()

parser.add_argument('--debug', action='store_true', help='Enable debug mode') parser.add_argument('--
video_path', type=str, default='videos/2_poland.mp4', help='Path to the video ') parser.add_argument('--lanes',
type=str, default=None, help='Lanes config')

args = parser.parse_args()

is_debug = args.debug video_path = args.video_path lanes = args.lanes

if lanes is None:
lanes = [DEFAULT_LANE_1, DEFAULT_LANE_2]
else:
lanes = [Lane(lane) for lane in json.loads(lanes)]

model_path = "mtlcr/models/road_segmentation.h5"

video_processor = VideoProcessor(model_path, debug=is_debug)
video_processor.process_video(video_path, lanes, 2)
```

Лістинг В.2. – Модуль video_processor.py

```
from datetime import datetime import json
import time
from uuid import uuid4

import cv2 import tensorflow

from mtlcr import calc_mtlcr, save_imgs tensorflow.keras.utils.disable_interactive_logging()

def create_mask(mask):
pred_mask = tensorflow.argmax(mask, axis=-1) pred_mask = pred_mask[..., tensorflow.newaxis] return pred_mask[0]



def preprocess_frame(frame):
return cv2.resize(frame, (256, 256))



class VideoProcessor:



def

init (self, model_path, debug=False):

self.model = tensorflow.keras.models.load_model(model_path) self.debug = debug

# first - vertical
# second - horizontal
# 0 - left high corner
def process_video(self, video_path, lanes, interval=10, simulation=True): cap = cv2.VideoCapture(video_path)

frame_rate = cap.get(cv2.CAP_PROP_FPS)
width = cap.get(cv2.CAP_PROP_FRAME_WIDTH) height = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)

frames_interval = round(intervalframe_rate) video = {

'id': uuid4(), 'path': video_path,
```

```
'frames_interval': frames_interval, 'interval': interval,
'width': width, 'height': height, 'lanes': [],
}

for i, lane in enumerate(lanes): lane_obj = {
'id': lane.id,
'n': i,
'coords': {
'dl': lane.coords[0],
'dr': lane.coords[1],
'hl': lane.coords[2],
'hr': lane.coords[3],
}
}

video['lanes'].append(lane_obj)

skip_counter = 0 while cap.isOpened():
# to debug on videos if simulation:
time.sleep(1 / frame_rate)

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

if skip_counter > 0:
skip_counter -= 1 continue

skip_counter = frames_interval self.process_frame(frame, video)
cap.release()


def process_frame(self, frame, video_metadata): preprocessed_frame = preprocess_frame(frame)
mask = self.model.predict(preprocessed_frame[tensorflow.newaxis, ...])

# Invert to make everything that is NOT road to be 1 mask = 1 - create_mask(mask).numpy()

results = []
time = datetime.utcnow().isoformat() for area in video_metadata['lanes']:
mtlcr, masks = calc_mtlcr(mask, video_metadata['width'], video_metadata['height'], area['coords']) res = {
'video': video_metadata['path'], 'lane_id': area['id'],
'mtlcr': mtlcr, 'created_at': time,
}
```

if self.debug:

folder_name = f"video_{video_metadata['id']}" timestamp = datetime.utcnow().isoformat() image_name = f"lane_{area['id']}_{timestamp}.png"

save_imgs(masks, f"MTLCR: {mtlcr}", folder_name, image_name) print(json.dumps(res))

results.append(res) return results

Лістинг В.3. – Модуль mtlcr.py

import os

import PIL import cv2

import numpy as np

from matplotlib import pyplot as plt

from matplotlib.backends.backend_agg import FigureCanvasAgg

SEGMENTATION_WIDTH = 256

SEGMENTATION_HEIGHT = 256

def calc_mtlcr(mask, init_width, init_height, corner_coords, threshold=25): reduced_corner_coords = reduce_corner_coords(corner_coords, init_width, init_height) transformed_mask = transform_perspective(mask, reduced_corner_coords) reduced_mask = reduce_transformed_mask(transformed_mask, threshold)

mtlcr = calc_mtlcr_by_reduced_mask(reduced_mask) return mtlcr, [mask, transformed_mask, reduced_mask]

def transform_perspective(img, corner_coords):src_points = np.array(corner_coords, dtype=np.float32)

target_width = corner_coords[1][0] - corner_coords[0][0] target_height = corner_coords[0][1] - corner_coords[2][1]

dst_points = np.array([

[0, target_height], [target_width, target_height], [0, 0], [target_width, 0]

], dtype=np.float32)

matrix = cv2.getPerspectiveTransform(src_points, dst_points)

result = cv2.warpPerspective(img.astype(np.uint8), matrix, (target_width, target_height)) result = result[:, :, np.newaxis]

return result

```
def reduce_transformed_mask(mask, threshold): car_pixels_count = np.sum((mask == [1]).all(axis=2), axis=1) car_pixel_percentage = car_pixels_count / mask.shape[1]100

mask = car_pixel_percentage > threshold
result = np.where(mask[:, None], np.array([1]), np.array([0])) return result

def reduce_corner_coords(corner_coords, init_width, init_height): return [
conv_point(corner_coords['dl'], init_width, init_height), conv_point(corner_coords['dr'], init_width, init_height), conv_point(corner_coords['hl'], init_width, init_height), conv_point(corner_coords['hr'], init_width, init_height),
]

def conv_point(point, init_width, init_height): return [
round(point[0] / init_widthSEGMENTATION_WIDTH), round(point[1] / init_heightSEGMENTATION_HEIGHT),
]

defcalc_mtlcr_by_reduced_mask(reduced_mask): total_pixels = reduced_mask.shape[0]
car_pixels = np.sum((reduced_mask == [1]).all(axis=1))

mtlcr = round(car_pixels / total_pixels, 4) return mtlcr

def get_mtlcr_plot_img(imgs, title): num_imgs = len(imgs)
fig, axes = plt.subplots(1, num_imgs, figsize=(15, 5)) # 1 row, num_imgs columns fig.suptitle(title, fontsize=16)

for i in range(num_imgs): axes[i].imshow(imgs[i], cmap='viridis') axes[i].axis('off') # Turn off axis labels axes[i].set_title(f'Image {i+1}')
plt.tight_layout(rect=[0, 0, 1, 0.95]) fig = plt.gcf()
canvas = FigureCanvasAgg(fig)
canvas.draw()
plot_image = np.array(canvas.renderer.buffer_rgba()) img = PIL.Image.fromarray(np.uint8(plot_image))

return img

def show_imgs(imgs, title):
img = get_mtlcr_plot_img(imgs, title) img.show()
```

def save_imgs(imgs, title, folder_name, file_name): img = get_mtlcr_plot_img(imgs, title) debug_folder = os.path.join("debug", folder_name) os.makedirs(debug_folder, exist_ok=True) img_path = os.path.join(debug_folder, file_name) img.save(img_path)

Лістинг В.4. – Модуль speed_tracker.py

import json import logging

import cv2
import numpy as np import pandas as pdfrom torch import Tensor
from ultralytics import YOLO from utils import get_rect_center
FRAMES_INTERVAL = 2
PREPROCESSED_VIDEO_WIDTH = 1080
PREPROCESSED_VIDEO_HEIGHT = 720
CROSSING_DETECTION_OFFSET = 20

class CrossingData:

def

init (self, lane_id, start, is_up):

self.lane_id = lane_id self.start = start self.processed = False self.is_up = is_up

class SpeedData:

def

init (self, lane_id, speed, start, finish, is_up):

self.lane_id = lane_id self.speed = speed self.start = start self.finish = finish self.is_up = is_up

class SpeedTracker:

```
def

init (self, model_path, class_list, object_tracker, lane_locator, debug=False):

self.model = YOLO(model_path) self.class_list = class_list self.object_tracker = object_tracker self.lane_locator = lane_locator self.object_id_2_crossing_data = {} self.lane_id_2_speed_data = {}for lane in self.lane_locator.lanes:
self.lane_id_2_speed_data[lane.id] = [] self.debug = debug

def process_video(self, video_path): cap = cv2.VideoCapture(video_path)
frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) fps = int(cap.get(cv2.CAP_PROP_FPS))

self.lane_locator.conv_lanes_coordinates(frame_width, frame_height, PREPROCESSED_VIDEO_WIDTH,
PREPROCESSED_VIDEO_HEIGHT)

frames_count = 0 while True:
ret, frame = cap.read() if not ret:
break

frames_count += 1
if frames_count % FRAMES_INTERVAL != 0: continue


frame = cv2.resize(frame, (PREPROCESSED_VIDEO_WIDTH, PREPROCESSED_VIDEO_HEIGHT))

results = self.model.predict(frame, device="mps", verbose=False) bounding_boxes = self.get_bounding_boxes(results)
bbox_id = self.update_tracker(bounding_boxes)

for bbox in bbox_id:
x3, y3, x4, y4, id = bbox
cx, cy = get_rect_center(x3, y3, x4, y4)

speed = self.process_speed(id, cx, cy, frames_count, fps) if speed is not None:
print(json.dumps(vars(speed)))
draw_elements(frame, bbox, cx, cy, id, x3, y3, x4, y4, speed) if self.debug:
self.draw_lanes_on_video(frame)

cv2.imshow("RGB", frame)
```

```
if cv2.waitKey(1) & 0xFF == 27: break

cap.release() cv2.destroyAllWindows()

def get_bounding_boxes(self, yolo_results): bounding_boxes = []
a = yolo_results[0].boxes.data
px = pd.DataFrame(Tensor.cpu(a)).astype("float")

for index, row in px.iterrows():
x1, y1, x2, y2, _, d = map(int, row[:6]) c = self.class_list[d]

vehicle_types = ['car', 'bus', 'truck', 'motorcycle'] if c in vehicle_types:
bounding_boxes.append([x1, y1, x2, y2]) return bounding_boxes



def update_tracker(self, bounding_boxes):
returnself.object_tracker.update(bounding_boxes)

def process_speed(self, id, cx, cy, frame_number, fps): lane = self.lane_locator.get_lane(cx, cy)
frame_time = frame_number / fps # time in seconds from start of the video.py crossing_data =
self.object_id_2_crossing_data.get(id)

speed_data = None
if crossing_data is not None and crossing_data.processed: return None

if lane is not None:
is_near_upper_boundary = lane.point_near_upper_boundary(cx, cy, CROSSING_DETECTION_OFFSET)
is_near_lower_boundary = lane.point_near_lower_boundary(cx, cy, CROSSING_DETECTION_OFFSET)
if crossing_data is None:
if is_near_upper_boundary:
self.object_id_2_crossing_data[id] = CrossingData(lane.id, frame_time, False) elif
is_near_lower_boundary:
self.object_id_2_crossing_data[id] = CrossingData(lane.id, frame_time, True) elif (crossing_data.is_up and
is_near_upper_boundary) or (
not crossing_data.is_up and is_near_lower_boundary): speed = (lane.length3.6) / (frame_time -
crossing_data.start)
speed_data = SpeedData(lane.id, round(speed), round(crossing_data.start, 2), round(frame_time, 2),
crossing_data.is_up)
self.lane_id_2_speed_data[lane.id].append(speed_data) crossing_data.processed = False

return speed_data
```

def draw_lanes_on_video(self, frame): for lane in self.lane_locator.lanes:

coords = np.array([lane.coords[0], lane.coords[1], lane.coords[3], lane.coords[2]], np.int32) cv2.polylines(frame, [coords], isClosed=True, color=(0, 255, 0), thickness=2)

if cv2.waitKey(1) & 0xFF == ord('q'): break

Лістинг В.4. – Модуль processing_service.py

import asyncio import json import time import uuid

from aiohttp import web from datetime import datetime

from app.utils import serialize_date_times, log, proc_type_2_short from tlir.tlir import calc_tlir

TLIR_CALCULATION = 'TLIR_CALCULATION' SPEED_EVALUATION = 'SPEED_EVALUATION' MTLCR_CALCULATION = 'MTLCR_CALCULATION'

class ProcessingService:
def
init (self, db_service, debug=False, log_levels=None):

if log_levels is None: log_levels = {}
self.active_processes = {} self.db_service = db_service self.debug = debug self.log_levels = log_levels

async def start_processing(self, request): video_id = request.match_info.get('video_id')
video = await self.db_service.get_video(video_id) if video is None:
return web.json_response(status=404)
parent_process_id = str(uuid.uuid4()) speed_process_id = str(uuid.uuid4()) tlir_process_id = str(uuid.uuid4())
asyncio.create_task(self.start_speed_calc_process(video, parent_process_id, speed_process_id)) mtlcr_process_id = str(uuid.uuid4())
asyncio.create_task(self.start_mtlcr_calc_process(video, parent_process_id, mtlcr_process_id)) asyncio.create_task(self.start_tlir_calc_process(video, parent_process_id, mtlcr_process_id,
tlir_process_id))
self.active_processes[speed_process_id] = {'process_type': SPEED_EVALUATION, 'video_path': video['link']}

self.active_processes[mtlcr_process_id] = {'process_type': MTLCR_CALCULATION, 'video_path': video['link']}

message = f"Processing started for {video['link']} with ID {parent_process_id} and subprocess IDs

```
        {speed_process_id} (SPEED_EVALUATION) and {mtlcr_process_id} (MTLCR_CALCULATION)"
log(message)

        return web.Response(text=message)

        async def start_speed_calc_process(self, video, parent_process_id, process_id): script_args =
['speed_tracker/tracker.py', '--video_path', video['link'], '--lanes',
        json.dumps(video['lanes'])]
        await self.start_external_process(parent_process_id, process_id, SPEED_EVALUATION, video['link'],
script_args, self.db_service.insert_speed_result)

        async def start_mtlcr_calc_process(self, video, parent_process_id, process_id):
        script_args = ['mtlcr/run_mtlcr.py', '--video_path', video['link'], '--lanes', json.dumps(video['lanes'])] await
self.start_external_process(parent_process_id, process_id, MTLCR_CALCULATION,
        video['link'], script_args, self.db_service.insert_mtlcr_result)

        async def start_external_process(self, parent_process_id, process_id, process_type, video_link, script_args,
result_saver): # process_id, timestamp, result
        await self.db_service.insert_active_process(parent_process_id, process_id, video_link, process_type)

        if self.debug: script_args.append("--debug")

        process = await asyncio.create_subprocess_exec( 'python3',
        script_args, stdout=asyncio.subprocess.PIPE, stderr=asyncio.subprocess.PIPE
        )

        while True:
        line = (await process.stdout.readline()).decode('utf-8') if not line:
        break

        if line == '\n' or line[0] != "{": continue

        ifself.log_levels[proc_type_2_short(process_type)]:
        log(f"New value from process {process_id} ({process_type}): {line}")

        result = json.loads(line)
        await result_saver(parent_process_id, process_id, datetime.utcnow(), result)

        if process_id not in self.active_processes: process.terminate()
        await process.communicate() break
```

```
await self.db_service.finish_active_process(process_id) log(f"Process {process_id} ({process_type}) finished")

del self.active_processes[process_id]

async def start_tlir_calc_process(self, video, parent_process_id, mtlcr_process_id, tlir_process_id, calc_interval=5, speed_calc_timeout=60): # calc_interval - seconds,
speed_calc_timeout - seconds while True:
time.sleep(calc_interval) for lane in video['lanes']:
mtlcr_list = await self.db_service.list_mtlcr_results_by_process_and_lane_id(parent_process_id, lane['id'], limit=1)
if len(mtlcr_list) == 0:
mtlcr = 0 else:
mtlcr = mtlcr_list[0]['result']['mtlcr']

speed_list = await self.db_service.list_speed_results_by_process_and_lane_id(parent_process_id, lane['id'], newer_than_seconds=speed_calc_timeout, limit=10)
max_speed = lane['max_speed']

speed_list = list(map(lambda obj: obj['result']['speed'], speed_list)) tlir = calc_tlir(mtlcr, speed_list, max_speed)
result = {'mtlcr': mtlcr, 'tlir': tlir}

result_log = {'video_id': video['id'], 'lane_id': lane['id'], 'mtlcr': mtlcr, 'tlir': tlir} log(f"New value from process {tlir_process_id} ({TLIR_CALCULATION}):
{json.dumps(result_log)}")

await self.db_service.insert_composed_result(video['id'], lane['id'], parent_process_id, datetime.utcnow(), result)

if mtlcr_process_id not in self.active_processes:
log(f"Process {tlir_process_id} ({TLIR_CALCULATION}) finished") break



async def stop_processing(self, request):
process_id = request.match_info.get('process_id')# id of subprocess

if process_id in self.active_processes: del self.active_processes[process_id]
return web.Response(text=f"Processing stopped for process ID {process_id}") async def list_processes(self, _request):
processes = await self.db_service.list_processes()
return web.json_response(processes)
```

```
        #----------Speed ----------

        async def list_speed_results_by_process_id(self, request): process_id = request.match_info.get('process_id')
        results = await self.db_service.list_speed_results_by_process_id(process_id) results = serialize_date_times(results)
        return web.json_response(results)

        async def list_speed_results_by_process_and_lane_id(self, request): process_id = request.match_info.get('process_id')
        lane_id = request.match_info.get('lane_id')

        results = await self.db_service.list_speed_results_by_process_and_lane_id(process_id, lane_id) results = serialize_date_times(results)
        return web.json_response(results)

        #----------MTLCR ----------

        async def list_mtlcr_results_by_process_id(self, request): process_id = request.match_info.get('process_id')
        results = await self.db_service.list_mtlcr_results_by_process_id(process_id) results = serialize_date_times(results)

        return web.json_response(results)

        async def list_mtlcr_results_by_process_and_lane_id(self, request): process_id = request.match_info.get('process_id')
        lane_id = request.match_info.get('lane_id')

        results = await self.db_service.list_mtlcr_results_by_process_and_lane_id(process_id, lane_id) results = serialize_date_times(results)
        return web.json_response(results)

        #----------MTLCR + TLIR ----------

        async def list_composed_results_by_process_id(self, request): process_id = request.match_info.get('process_id')
        results = await self.db_service.list_composed_result_by_process_id(process_id) results = serialize_date_times(results)
        return web.json_response(results)
```

ДОДАТОК Г

(обов'язковий).

Ілюстративна частина

ІЛЮСТРАТИВНА ЧАСТИНА

Автоматизована система моніторингу відеопотоку дорожнього перехрестя

Студент групи 2АКІТР-23м
Ярослав РОЇК
Підпис *(Ім'я та ПРІЗВИЩЕ)*

Керівник: Д.т.н., проф. КСУ
Марія ЮХИМЧУК
Підпис *(Ім'я та ПРІЗВИЩЕ)*

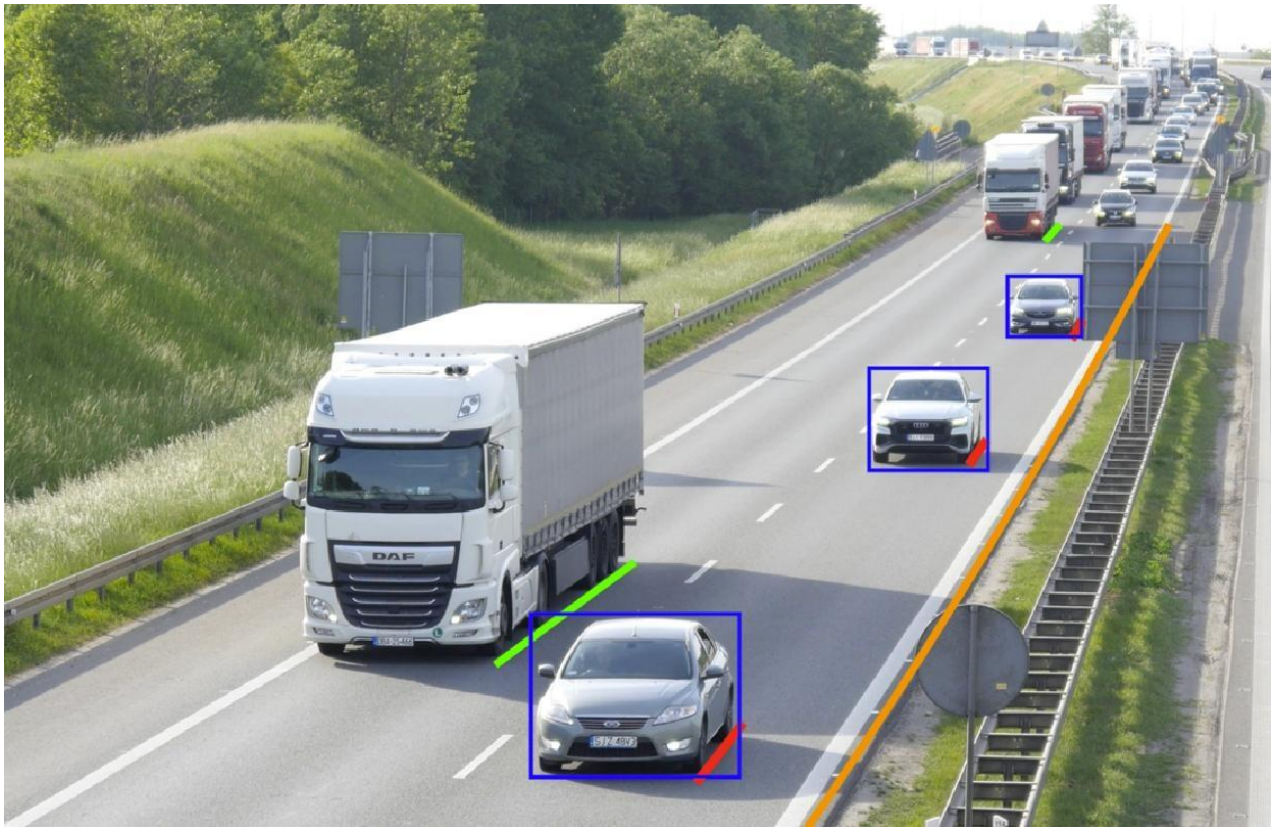

Рисунок І1-Кадр з камери відеоспостереження біля міста Вроцлав

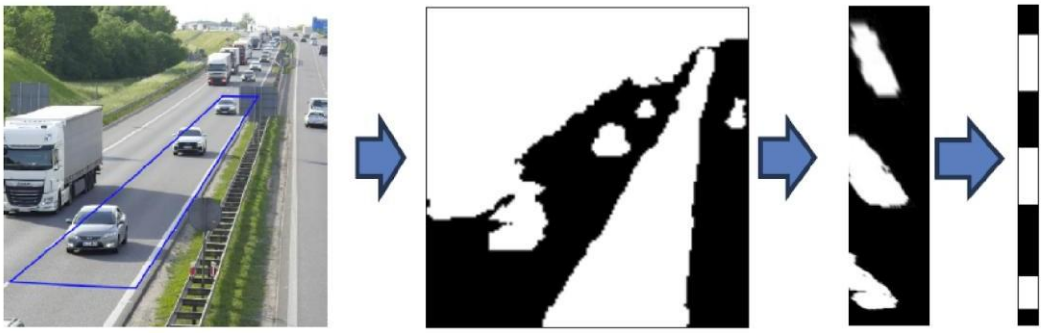

Рисунок І2-Перетворення зображень для розрахунку MTLCR

Рисунок І3-Візуалізація процесу розрахунку MTLCR

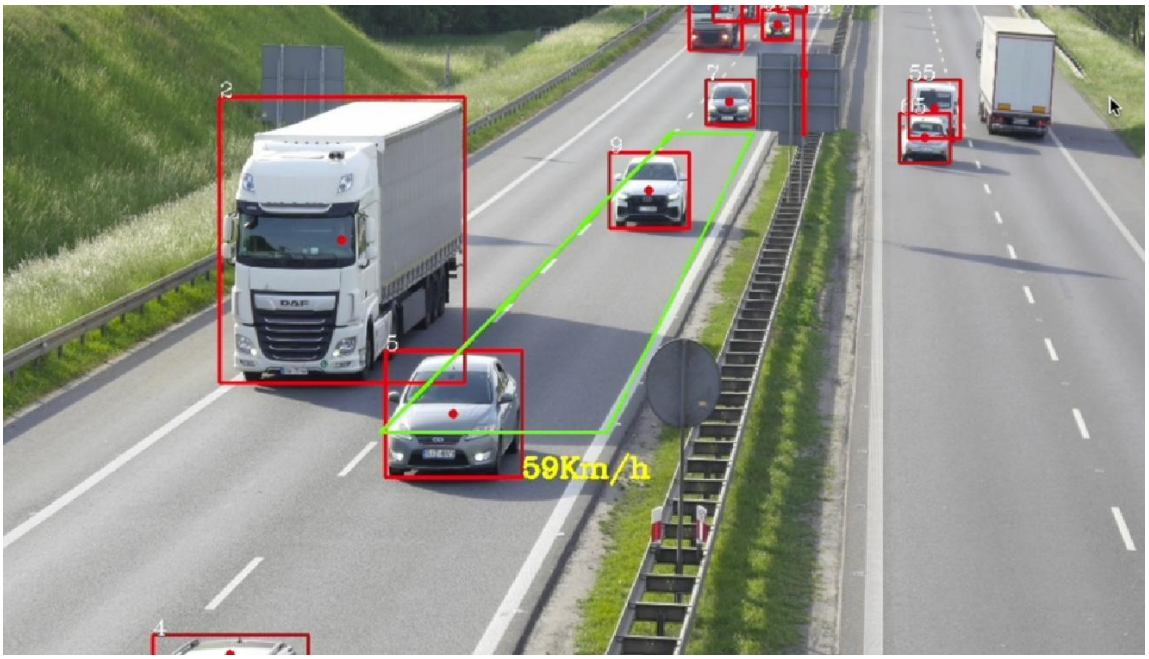

Рисунок І4-Візуалізація процесу оцінки швидкості транспортних засобів

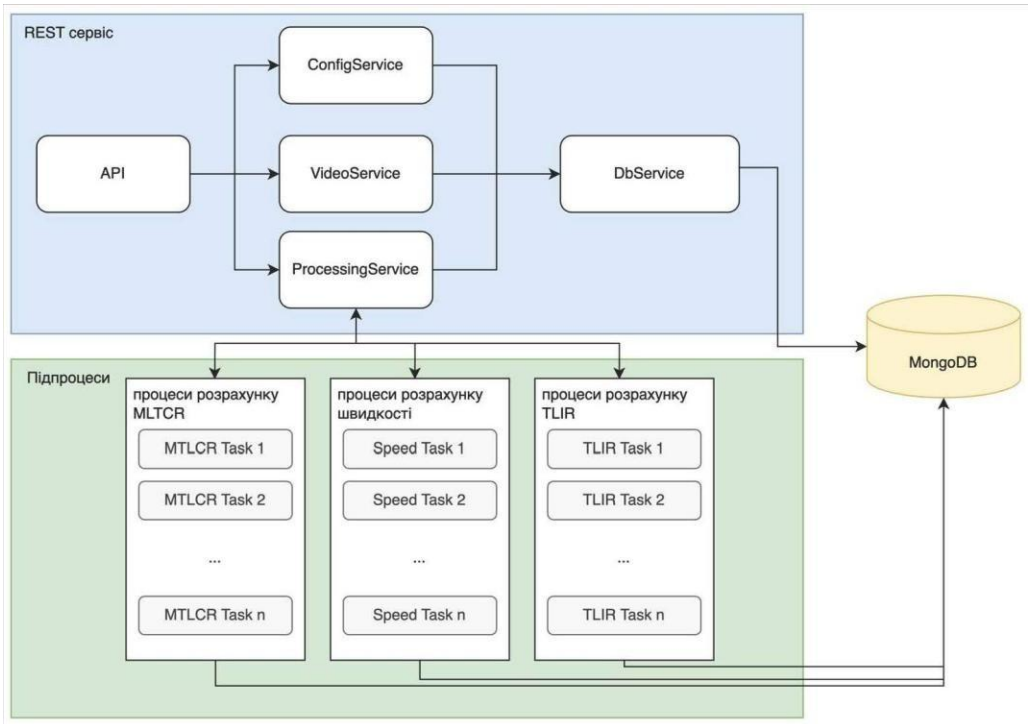

Рисунок І5-Архітектура програмного засобу

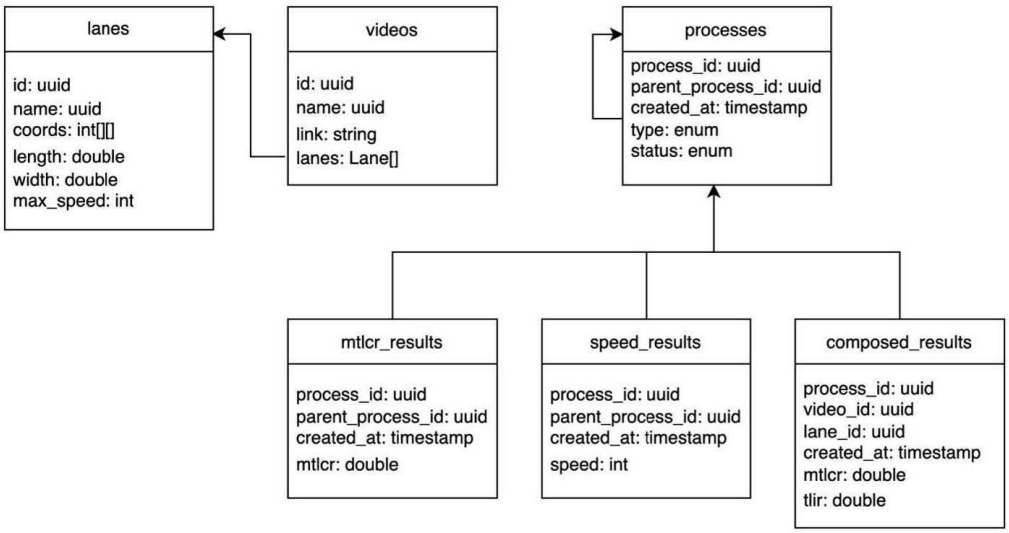

Рисунок І6-Схема бази даних

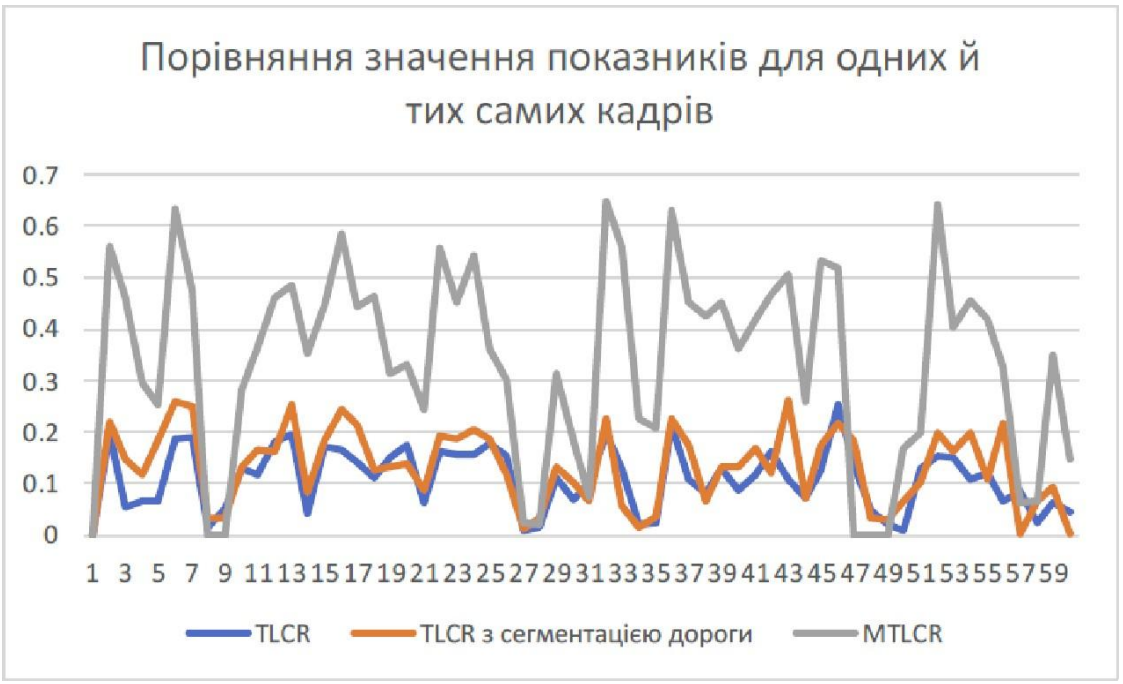

Рисунок І7-Порівняння обрахованих показників для одних й тих самих кадрів

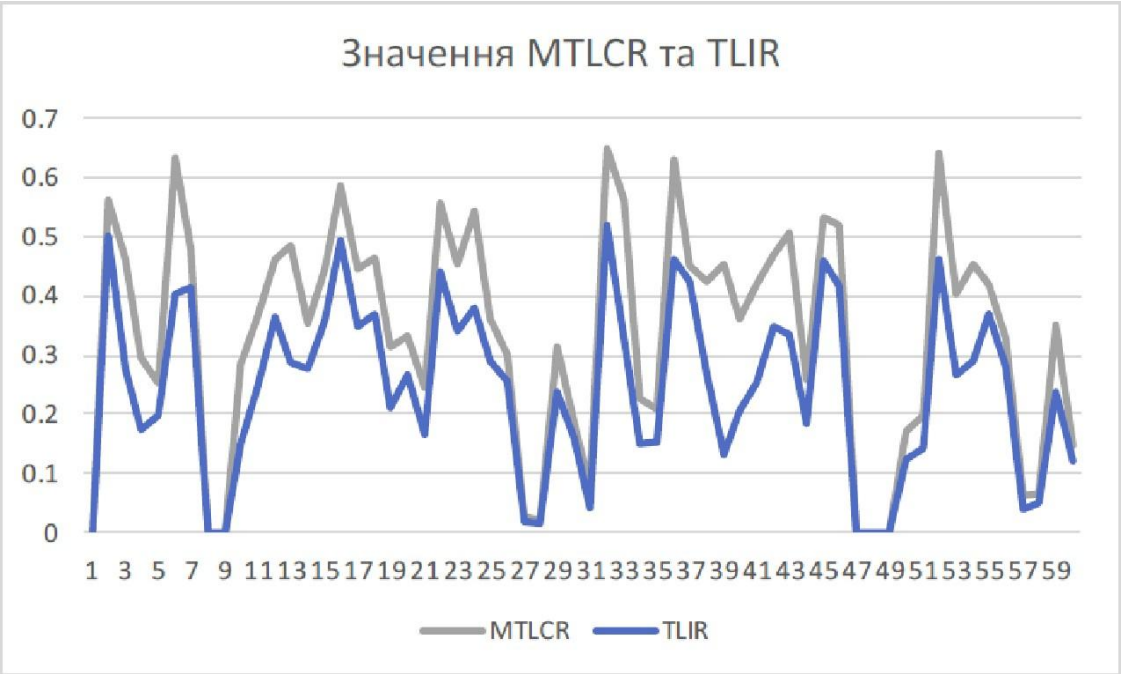

Рисунок І8-Візуалізація розрахованої системи показників